



Universidad
Carlos III de Madrid
www.uc3m.es

Grado en Ingeniería en
Tecnologías de Telecomunicación

Trabajo Fin de Grado

Estudio de mecanismos de distribución de contenidos en redes vehiculares urbanas

Juan José Campaña Montes

Tutor

Ignacio Soto Campos

Leganés, Junio 2015

Nota aclaratoria

De acuerdo con la normativa del Trabajo de Fin de Grado para el plan 2011 en el Grado en Ingeniería en Tecnologías de Telecomunicación, se incluyen en este proyecto en inglés los apartados introducción (página 11) y conclusiones (página 65), además de un resumen extendido en dicho idioma incluido como anexo al final del documento (página 83).

Índice de capítulos

Índice de capítulos	3
Índice de ilustraciones	5
Índice de tablas	6
Agradecimientos	7
Resumen	8
Abstract.....	8
1. Introduction/Introducción.....	9
1.1. Versión en español.....	9
1.2. English version	11
2. Estado del arte	13
2.1. Introducción a las redes vehiculares	13
2.2. Tipos de redes vehiculares	15
2.2.1. VANET.....	15
2.2.2. DTN	16
2.2.3. VDTN	17
2.3. Aplicaciones.....	18
2.3.1. Seguridad vial	18
2.3.2. Eficiencia y gestión del tráfico	20
2.3.3. Aplicaciones de información y entretenimiento	20
2.4. Regulación actual.....	21
2.4.1. Estandarización en Estados Unidos	21
2.4.2. Estandarización en Japón.....	22
2.4.3. Estandarización en la Unión Europea.....	23
2.5. Entorno socio-económico	23
2.6. Protocolos de distribución de contenidos en VANETs	24
2.7. Simulación basada en agentes.....	30
3. Distribución de contenidos: Procedimiento	34
3.1. Mapa	34
3.2. Programas utilizados.....	35
3.2.1. SUMO.....	35
3.2.2. TRACI4J.....	36
3.2.3. MASON.....	37

3.2.4.	Obstáculos	38
3.2.5.	Diagrama general.....	39
3.3.	Estructura interna y agentes	40
3.4.	Consideraciones sobre los protocolos usados	41
4.	Resultados de las simulaciones	48
4.1.	Densidad del tráfico	48
4.2.	Número de contenidos	50
4.3.	Popularidad	53
4.4.	Distribución de contenidos tipo Internet.....	56
5.	Alternativas de diseño	59
5.1.	Simulación en otras plataformas	59
5.2.	Factores no tenidos en cuenta	60
6.	Conclusions/Conclusiones	62
6.1.	Versión en español.....	62
6.1.1.	Líneas futuras de investigación	64
6.2.	English version	65
6.2.1.	Future research	67
7.	Bibliografía.....	68
	Anexo I. Acrónimos y siglas	71
	Anexo II. Planificación y costes	72
	Planificación	72
	Costes y presupuesto.....	74
	Materiales necesarios	74
	Gastos de personal.....	75
	Costes indirectos	75
	Presupuesto detallado.....	75
	Anexo III. Manual de instalación del entorno de simulación.....	77
	Appendix IV. Extended abstract.....	83

Índice de ilustraciones

Ilustración 2-1. Ejemplo de red vehicular para avisar a las autoridades de una emergencia	14
Ilustración 2-2. Funcionamiento de los protocolos tipo Push	25
Ilustración 2-3. Funcionamiento de los protocolos tipo Pull.....	26
Ilustración 2-4. Atributos y métodos de un agente vehículo	31
Ilustración 2-5. Comunicación entre agentes a través de sus métodos y cambio de sus atributos.	32
Ilustración 2-6. Interacción de los agentes con el medio en el que se encuentran	33
Ilustración 3-1. Plano de Madrid utilizado, de 4.5 km x 4.5 km	35
Ilustración 3-2 Plano de Madrid visto en MASON (Izquierda) y plano aumentado mostrando los agentes (derecha).....	37
Ilustración 3-3. Selección de datos de un vehículo	38
Ilustración 3-4. Componentes del entorno de simulación	39
Ilustración 3-5. Menú de selección de parámetros	40
Ilustración 3-6. Funcionamiento de Push	42
Ilustración 3-7. Funcionamiento de Push optimizado	43
Ilustración 3-8. Funcionamiento de Pull.....	44
Ilustración 3-9. Funcionamiento de Pull optimizado.....	44
Ilustración 3-10. Funcionamiento de MHTI.....	45
Ilustración 3-11. Funcionamiento de MHTI optimizado.....	46
Ilustración 4-1. Tasa de éxito variando densidad de tráfico (1 contenido).....	48
Ilustración 4-2. Tasa de éxito con múltiples contenidos (6000 vehículos)	51
Ilustración 4-3. Tasa de éxito con contenidos con popularidad (6000 vehículos y 1000 contenidos)	54
Ilustración 4-4. Tasa de éxito con $\alpha = 0.8$ y 1000 contenidos	57
Ilustración 0-1. Diagrama de Gantt	74

Índice de tablas

Tabla 4-1 Sobrecarga de la red variando la densidad de tráfico.....	49
Tabla 4-2. Sobrecarga de la red variando el número de contenidos.....	52
Tabla 4-3. Media de contenidos almacenados.....	52
Tabla 4-4. Sobrecarga de la red con contenidos con popularidad	55
Tabla 4-5. Media de contenidos con popularidad.....	56
Tabla 4-6. Sobrecarga de la red con una distribución de contenidos similar a la de Internet	57
Tabla 4-7. Contenidos almacenados en un modelo similar al de Internet.....	58

Agradecimientos

El desarrollo de este trabajo no habría sido posible sin la ayuda y la colaboración de Ignacio Soto, el cual ha realizado un seguimiento continuo en el desarrollo del mismo, y ha estado en todo momento dispuesto a resolver cualquier problema que se le planteara. Agradecer también a los profesores Manuel Urueña y María Calderón por proporcionar tanto información como material necesario para poder avanzar en puntos clave del proyecto.

Gracias también al departamento de Ingeniería Telemática de la Universidad Carlos III por facilitarme un laboratorio con el material necesario para poder desarrollar el entorno de simulación, y al equipo de técnicos que han prestado su ayuda ante cualquier problema que ha surgido.

Por último, agradecer a mi familia y a mi pareja todo el apoyo que me han brindado durante la realización del proyecto, y el interés que han mostrado por conocer cuál era el estado del mismo.

Resumen

El objetivo de este trabajo es la investigación, usando un modelo basado en agentes, de formas eficientes de distribuir contenidos relacionados con la información y el entretenimiento en una red formada por vehículos. Las simulaciones basadas en agentes permiten estudiar el comportamiento resultante de interacciones complejas entre entidades autónomas, por lo que este trabajo se ha creado como un punto de partida para otros trabajos futuros relacionados con redes vehiculares que no puedan realizarse fácilmente con simuladores de comunicaciones convencionales.

Diferentes protocolos se han propuesto durante los últimos años para distribuir contenidos en una red vehicular. En este trabajo se utilizará un modelo basado en agentes para simular una situación real en la ciudad de Madrid, donde los coches se moverán durante 3600 segundos y estarán interesados en obtener un único contenido. Su objetivo es conseguirlo usando la red vehicular, pero si no lo consiguen, usarán una red celular para descargarlo.

Durante la simulación, se variarán parámetros como la densidad de tráfico, el número de contenidos en la red y la popularidad de estos contenidos (algunos contenidos son más interesantes para los usuarios que otros). Tres protocolos (cada uno con posibilidad de añadir optimizaciones) serán probados en cada situación, y los resultados se compararán para decidir qué protocolo es el mejor en cada escenario.

Abstract

The purpose of this work is to research, using an agent based model, about efficient ways of distributing infotainment contents in a network formed by vehicles. Agent based simulations can deal with complex behaviors resulting from the interaction of autonomous entities, so this work intends to be an initial step towards future research about vehicular networks which cannot be implemented easily by conventional communications simulators.

Different kinds of protocols have been proposed during the past years to distribute contents using a vehicular network. In this work, we use an agent-based model to simulate a real situation in the city of Madrid, where cars travel during 3600 seconds and they are be interested in obtaining a single content. Their aim is to get the content using the vehicular network but, if that is not possible, they use a cellular network to download it.

During the simulations, we study the effect of parameters such as traffic density, number of contents in the network and popularity of these contents (some contents are more interesting to users than others). Three different protocols (each one with possible optimizations) are tested in the different scenarios to evaluate their performance.

1. Introduction/Introducción

1.1. Versión en español

Tecnología y comunicaciones han estado evolucionando durante los últimos años para permitir la transmisión de información en cualquier sitio. Ordenadores portátiles, teléfonos inteligentes y otros dispositivos necesitan estar conectados a Internet incluso cuando están en movimiento, por lo que tienen que usar tecnologías inalámbricas como 3G o 4G para ello. El incremento del número de usuarios que demandan acceso móvil a Internet y la aparición de nuevos escenarios de aplicación, como es el hecho de conectar vehículos, requiere soluciones basadas en redes inalámbricas que provean conectividad a los mismos de forma eficiente.

El propósito de conectar vehículos a Internet es dar a los usuarios nuevos servicios que antes no podían disfrutar, como información a tiempo real del tráfico para evitar congestión en las carreteras, servicios de emergencia como el envío de mensajes a un hospital para pedir una ambulancia tras un accidente, o la posibilidad de descargar contenidos, tanto audio como vídeo. También existen oportunidades de negocio en las redes vehiculares. Por ejemplo, anuncios publicitarios pueden ser enviados a zonas geográficas específicas para que sean visualizados por personas conduciendo en los alrededores. Se pueden clasificar los servicios que proporcionan las redes vehiculares en tres categorías: Seguridad vial, gestión de la eficiencia del tráfico, y servicios de información y entretenimiento. En este trabajo, se analizará la distribución de contenidos en redes vehiculares en entornos urbanos para aplicaciones no relacionadas con la seguridad.

En escenarios urbanos, sobre todo grandes ciudades y metrópolis, el uso exclusivo de tecnología celular como 3G o 4G podría ser ineficiente debido al dinamismo de la red, la aparición de túneles o la cantidad de tráfico de datos que generarían tanto los vehículos como los usuarios móviles, sobrecargando la red. Es por eso que se requiere otra solución distinta. Si los vehículos pudieran comunicarse los unos con los otros y cooperar para hacer una distribución de contenidos efectiva, la red 3G tendría que manejar menos tráfico. Esta forma de comunicación se conoce por su definición anglosajona: Vehicular Ad Hoc Network (VANET)¹.

Se han sucedido numerosas teorías sobre cómo la comunicación podría ser más efectiva, es por eso que en el capítulo 2 (“Estado del arte”) se mencionarán algunas de ellas, explicando sus ventajas y sus inconvenientes. Este trabajo se basa en el uso de VANET teniendo en cuenta que los vehículos pueden almacenar y transportar sus contenidos, por lo que las transmisiones no son instantáneas. Por otro lado, los vehículos no pueden esperar demasiado tiempo para recibir un contenido, porque el interés por el mismo puede disminuir o incluso desaparecer. Es por eso que en el sistema usado en este proyecto, los

¹ Todas las siglas y acrónimos y su significado correspondiente pueden encontrarse al final del documento en el Anexo I. Acrónimos y siglas.

vehículos intenta en primera instancia conseguir el contenido a través de la VANET, pero después de un intervalo de tiempo sin conseguirlo, descargan el contenido usando la red celular.

El caso de estudio de esta investigación es analizar el comportamiento de la VANET utilizando diferentes protocolos de distribución de contenidos. Dos grandes tipos pueden distinguirse, protocolos de tipo Push, los cuales periódicamente envían un contenido aleatorio en broadcast, y los protocolos de tipo Pull, que periódicamente piden un contenido a vehículos cercanos. Otro protocolo estudiado es MHTI (Multi-Hop To Infostation) cuya diferencia con Pull es que intenta llegar a una estación de información² a través de varios saltos para obtener el contenido. Cada protocolo puede estar implementado con optimizaciones que mejoran su comportamiento en la red.

Los parámetros que se han usado para comparar cada protocolo son la tasa de éxito de obtener un contenido a través de la VANET y la sobrecarga en la red que se ha generado, variando la densidad del tráfico urbano (la VANET dará mejores resultados cuando las carreteras estén repletas de vehículos), la cantidad de contenidos que pueden aparecer, y la frecuencia de aparición de cada uno, es decir, si existe un nivel de popularidad para los contenidos.

Se ha utilizado un modelo de simulación basado en agentes para recrear el comportamiento de cada vehículo, desplazándose a través de las calles y cooperando para hacer que la red funcione. No se han considerado posibles nodos egoístas o maliciosos que podrían no ser cooperativos, por lo que todos los vehículos siguen estrictamente el protocolo con el fin de simplificar la simulación.

El movimiento de los vehículos se ha generado automáticamente utilizando la herramienta SUMO (Simulator of Urban Mobility), utilizando el centro de Madrid como el escenario de la simulación. Las coordenadas de cada uno de los coches se envían a MASON (MultiAgent Simulator of Networks), el cual es el responsable de la inteligencia de la red. Controla cuándo tiene que transmitir cada vehículo y qué contenidos tiene que almacenar. Se consideran como obstáculos para la comunicación los edificios de la ciudad, por lo que si dos vehículos no tienen visión directa por la existencia de un edificio entre ellos, no se puede establecer satisfactoriamente dicha comunicación.

Tras esta introducción al proyecto, en el Capítulo 2 (“Estado del arte”) se realiza una panorámica de las redes vehiculares, sus distintos tipos, aplicaciones, los esfuerzos de distintos países y organizaciones por estandarizar este nuevo tipo de tecnología, y se mostrarán propuestas de protocolos de distribución de contenidos en la literatura. En el Capítulo 3 (“Distribución de contenidos: procedimiento”) se muestran los programas que componen el entorno de simulación que se ha creado, además de las características de los protocolos que se han utilizado para las simulaciones y el escenario en el que viajan y se comunican los vehículos que componen la VANET. El Capítulo 4 (“Resultados de las

² Una estación de información es un nodo fijo en la infraestructura (conectado a Internet) que también pertenece a la VANET.

simulaciones”) se muestra qué datos se han extraído de cada simulación en cada uno de los protocolos, destacando los puntos más representativos. En el Capítulo 5 (“Alternativas de diseño”) se mencionan otras posibilidades a la hora de afrontar la realización de un estudio sobre redes vehiculares, tanto a nivel de implementación como a nivel conceptual. Por último, se añade un capítulo de conclusiones donde se destacan los puntos principales que se pueden concluir del trabajo y las líneas futuras que se han de seguir.

1.2. English version

Technology and communications have been evolving during the recent years to permit the transmission of information everywhere. Laptops, smartphones and other devices need to be connected to Internet even when they are moving, so they have to use wireless technologies as 3G or 4G for this purpose. Increasing number of users demand mobile access and new application scenarios have appeared, such as connected vehicles, which requires wireless network solutions to efficiently provide connectivity to them.

The aim of connecting vehicles to Internet is to give users new services that they couldn’t enjoy before, as real-time traffic information to avoid road congestion, emergency services like sending a message to the hospital to ask for an ambulance because of an accident, or video and audio streaming. There is also a business opportunity in vehicular networks. For example, advertisements can be sent to specific geographical areas, to target people driving in the surroundings. We can classify services for vehicular networks in three categories: safety, traffic efficiency and infotainment (information and entertainment). In this work, we analyze content distribution solutions in vehicular networks for non-safety applications in urban scenarios.

In urban scenarios, big cities and metropolis overall, the exclusive use of 3G/4G technology could be inefficient because of the dynamism of the network, the apparition of tunnels or the amount of network traffic that such vehicles and other mobile users can generate, potentially overloading the network. Therefore, another network solution is needed. If cars communicate with each other and cooperate to make an effective content distribution, among them, the 3G network would manage less traffic. This way of communication is called Vehicular Ad Hoc Network (VANET)

There have been many theories about how this communication could be more effective, so in chapter 2 (“State of art”) some of them will be mentioned, explaining their advantages and drawbacks. This work is based on the use of VANET, having into account that vehicles can store and carry their contents, so transmissions are not instantaneous. On the other hand, vehicles can’t be waiting for a content too much time because the interest on this content can’t disappear or loose utility. That is why, in our system, vehicles try to get a content first through the VANET, but after a time interval without getting it, vehicles download the content using the 3G network.

The case of study of this research is to analyze the behavior of the VANET using different protocols for content distribution. Two main types can be distinguished, Push protocols, which periodically send one random content in broadcast, and Pull protocols, that periodically ask for a content to nearby vehicles. Another protocol studied is called MHTI (Multi-Hop To Infostation) whose difference with Pull is that it tries to access to an infostation³ to get the content in various hops. Each protocol has optimizations that improve their performance in the network.

The performance parameters used to compare each protocol are the success rate of getting contents using the VANET, and the overhead that is generated. We study the performance, varying the urban traffic density of vehicles (VANET will perform better when streets are crowded of vehicles), the amount of contents that can appear and the frequency of apparition of each content, i.e., the level of popularity for the contents.

An agent based simulation model is used to recreate the behavior of each vehicle, moving through the streets and cooperating to make the network work. We have not considered selfish or malicious nodes that could not be cooperative, so every vehicle follows strictly the protocol to simplify the simulation.

The movements of the vehicles have been automatically generated using SUMO (Simulator of Urban Mobility), using the center of Madrid as the simulation scenario. The coordinates of each car are sent to MASON (MultiAgent Simulator of Networks), which is responsible of the network intelligence. It controls when each vehicle has to transmit and which contents it has store. We consider the buildings of the city as obstacles to communications, so if two vehicles can't see each other because there is a building in the middle, they cannot establish successful communications.

After this brief introduction, in Chapter 2 ("State of art") is shown an overview of vehicular networks in general, different types, applications and standardization in the most important countries and organizations. We also review different content distribution protocols proposed in these years. In Chapter 3 ("Content distribution: Procedure") we explain the programs that make up the developed simulation environment, including the main features of each protocol used in the simulations and the scenario where vehicles are travelling and communicating. Chapter 4 ("Simulations results") shows data extracted from each simulation, focusing the analysis in the most surprising results. In Chapter 5 ("Design alternatives"), it is mentioned which other possibilities (conceptual and technical) could have been taken into account for this work. Finally, a chapter for conclusions has been included where main tasks are highlighted and future research is proposed.

³ An infostation is a static node in the infrastructure (connected to Internet) that belongs to the VANET.

2. Estado del arte

En este capítulo se realizará una panorámica sobre las redes vehiculares y la distribución de contenidos. Se mostrará el origen de estas redes, sus diferentes tipos, las aplicaciones que puede tener este tipo de tecnología y qué esfuerzos se han realizado para estandarizarla. Se hará especial hincapié en la explicación de protocolos de distribución de contenidos y de la simulación basada en agentes, pues son los dos pilares que componen esta investigación.

2.1. Introducción a las redes vehiculares

Desde la década de los 90, el paradigma habitual de lo que se conocía como una red de ordenadores ha ido evolucionando poco a poco. El motivo principal es el cambio en la infraestructura de la misma red. Normalmente, las redes estaban compuestas de dispositivos fijos, interconectados entre sí a través de redes de cable que permitían una conectividad estable la mayor parte del tiempo. La estructura general se podría considerar centralizada, en la que los dispositivos con más capacidad se situaban en el entramado central y se encargaban de proveer las peticiones de los ordenadores situados en el extremo de la red.

En cambio, la aparición de nuevos dispositivos como ordenadores portátiles, *smartphones* y *tablets* cambió radicalmente este modelo de redes tan estáticas, por lo que se empezó a desarrollar otro tipo de red llamada MANET (Mobile Ad-Hoc NETwork). Como se enuncia en [1], este tipo de redes aceptan una estructura descentralizada, así como un mayor dinamismo en el que los dispositivos pueden estar en movimiento, cambiar de punto de acceso a la red, etc., aunque la función principal sigue siendo proporcionar conectividad a Internet. Aprovechando su conexión inalámbrica, los dispositivos se conectan unos con otros y forman temporalmente una red en la que a pesar de su movilidad consiguen acceso a Internet a través de otros dispositivos [2].

A raíz de este tipo de redes surgió la idea de dotar de conectividad a un colectivo que hasta ese momento no estaba relacionado con las comunicaciones ni las redes: los vehículos. Durante todas las décadas de existencia del automóvil, estos se han concebido solamente como un medio de transporte únicamente controlado por la habilidad de la persona que lo conduce. En cambio, las elevadas cifras de siniestros en las carreteras, los grandes problemas de circulación de algunas ciudades o algunas autopistas y la posibilidad de disfrutar de contenidos o buscar información dentro del vehículo hicieron surgir la idea de dotar de capacidad de comunicación y acceso a la red a los mismos, formando lo que se llamaría una red vehicular, como se puede ver en la Ilustración 2-1.



Ilustración 2-1. Ejemplo de red vehicular para avisar a las autoridades de una emergencia

Las características que hacen distinta a una red vehicular de otro tipo de red móvil son esencialmente tres según [3]:

- **Energía de transmisión ilimitada:** Generalmente los dispositivos que forman una red móvil tienen una gran limitación de energía al tener simplemente una batería que los alimente sin posibilidad de carga en cualquier momento. En cambio, los vehículos pueden producir energía eléctrica a través del consumo de combustible, o incluso se puede potenciar la generación añadiendo paneles solares.
- **Gran capacidad de cómputo:** El tamaño de los vehículos hace que no suponga un problema la capacidad de cómputo para la transmisión o la lectura de sensores, mientras que en un dispositivo de un tamaño mucho más reducido es un factor determinante.
- **Movilidad predecible:** Al contrario que en las redes móviles tradicionales, la movilidad por carreteras es fácilmente predecible, ya que hay vías que están más concurridas, trayectos más probables o destinos frecuentados. Además, el hecho de que la movilidad esté limitada a las carreteras hace aún más restrictiva la red en el espacio, por lo que los movimientos a realizar son notablemente inferiores a los de un peatón por ejemplo, que puede entrar en edificios, o atravesar calles.

El principal problema que ofrece este tipo de redes es la alta movilidad de los nodos que componen la red, ya que en una MANET los nodos tienen la capacidad de no verse afectados por algunos cambios de posición, pero esta es limitada, y en ningún caso llega a ser tan intermitente y con tantos errores como en una red vehicular. En cambio, la alta velocidad a la que circulan los vehículos, además de la cantidad de obstáculos que se encuentran en una ciudad para establecer la comunicación hacen imposible la adopción de una red MANET tal y como han sido diseñadas como una forma de interconexión de vehículos.

También se puede pensar que sería útil la utilización de conexión celular como puede ser el uso de una red 3G o 4G en la que no sea necesario conectar varios vehículos entre sí si se puede tener conexión directa a Internet. La problemática de esta vía es que en el momento en que los vehículos se encuentren en zonas sin cobertura no podrán comunicarse en momentos de emergencia, y si todos los vehículos deciden descargar contenidos de un tamaño considerable, la red podría sufrir congestión porque los

proveedores de servicio tienen que atender una demanda cada vez mayor de usuarios móviles. Es por eso que se encuentra la necesidad de conectar los vehículos entre sí para aprovechar contenidos que ya se hayan conseguido en la zona previamente, aliviando así la red celular.

Por otro lado, la alta movilidad de los nodos de la red también ofrece la oportunidad de conseguir una gran difusión geográfica de los contenidos sin tener que volver a descargarlos de Internet, ya que un vehículo que posea un contenido puede compartirlo en varias zonas durante el tiempo en que está en movimiento, evitando que este se quede estancando en un único grupo de vehículos vecinos.

En una red vehicular los vehículos cooperarán entre sí para conseguir la información que deseen descargar. Esta información puede ser obtenida de algún vehículo cercano que tenga el contenido, o de las denominadas estaciones de información, las cuales son nodos fijos situados en las intersecciones más importantes de las vías por las que circulan los vehículos. Las estaciones tienen acceso a Internet y se encargan de proveer a los vehículos de la información que les soliciten. Sin la existencia de las estaciones de información los vehículos estarían obligados a implementar una conexión a la red celular y las primeras demandas de una información estarían obligadas a ser descargadas por 3G.

Para suplir estos déficits y aprovechar las oportunidades que presenta un escenario como el descrito, se han desarrollado las llamadas redes vehiculares, específicas para la interconexión de vehículos y que en los próximos apartados se analizarán con más detalle pues forman la base conceptual del proyecto que se ha realizado.

2.2. Tipos de redes vehiculares

En el apartado anterior se han enunciado cuáles eran los principales problemas que presentaban las redes vehiculares, causados por su propia naturaleza. Las soluciones que se han propuesto se pueden clasificar según [2] [4] en tres arquitecturas diferentes, VANET (Vehicular Ad-hoc NETwork), DTN (Delay Tolerant Network) y VDTN (Vehicular Delay Tolerant Network). A pesar de esta clasificación, el nombre VANET es el que se utiliza comúnmente para denominar todas las redes vehiculares. Durante el desarrollo de este documento, se realizará numerosas menciones a la red que se ha implementado como VANET, a pesar de que esta no representa exactamente esta arquitectura, sino que es una mezcla de varias.

2.2.1. VANET

Las denominadas VANET representan la adaptación directa de las MANET a las redes vehiculares, considerándose incluso una clase de MANET. En esta arquitectura los vehículos actúan como nodos y en cuanto se desea realizar una comunicación, se crea un camino con los nodos que están disponibles en ese instante y que participarán en la propagación del mensaje. La creación de estos caminos hace que este tipo de redes no funcionen bien con los cortes de comunicación ni con los retardos, pues un retardo puede hacer que la topología haya cambiado totalmente y ya no sea posible la utilización del

camino establecido. Las comunicaciones pueden ser desde un vehículo hasta otro, (V2V, Vehicle to Vehicle) o desde un vehículo a la estación que provea conexión a Internet (V2I, Vehicle to Infrastructure), en función de si el objetivo de la comunicación es enviar un mensaje a otro vehículo o comunicarse con nodos en la red fija (Internet).

Como se ha comentado, el adaptar una arquitectura tipo MANET a una topología formada por vehículos supone cambios en la topología demasiado rápidos y la necesidad de fragmentar el camino para poder responder a los mismos. Los protocolos tradicionales no son los adecuados para los vehículos, por lo que se han desarrollado otros que responden mejor a estos cambios de topología. El desarrollo de estos protocolos debe tener en cuenta que las aplicaciones en una VANET pueden ser de naturaleza muy distinta unas de otras, por lo que deben crearse varios protocolos que satisfagan todas las situaciones.

En función de las aplicaciones, será necesaria la transmisión en *unicast*, que implica transmitir desde un nodo hasta otro concreto, *multicast* o *geocast*, que implica la transmisión a varios vehículos seleccionados, ya sea por la zona geográfica o porque pertenecen a un grupo concreto, y el *broadcast*, que supone el envío a todos los vehículos que puedan recibir el mensaje sin ningún tipo de discriminación.

Este tipo de red no es la más adecuada para la distribución de contenidos, que es el propósito de este trabajo, ya que en este tipo de aplicación el tiempo no es tan crítico y no requieren la búsqueda de vehículos concretos, sino la diseminación de contenidos a la red para que cada vehículo obtenga los que desee. Las VANETs son más adecuadas para aplicaciones en las que sea crítico el tiempo de llegada hacia el destino, generando un camino hasta el destino. Un ejemplo de este tipo de aplicaciones pueden ser las de emergencias, en las que cada segundo de retardo puede ser esencial.

2.2.2. DTN

Las redes DTN surgieron para suplir las carencias que presentaban las VANETs ante cortes de conexión y retardos en el tiempo. Esto se realiza cambiando el papel que desempeñan los vehículos en la red, no siendo únicamente nodos que retransmiten inmediatamente un contenido hasta un destino, sino que puede recibir un dato, almacenarlo y al llegar a una posición donde haya posibilidad de enviar a otro vehículo, realiza la transmisión siguiendo una ruta determinada para alcanzar el destino.

Las redes tradicionales no permiten realizar modelos como este pues están enfocadas “extremo a extremo” y requieren una conexión permanente. Si uno de los nodos de la conexión se desconecta de la red, y no hay otro camino que tomar en ese momento, se da por imposible la conexión y el nodo origen tiene que reintentar de nuevo. En cambio, las redes DTN proponen un sistema “salto a salto”, en el cual se van creando sesiones en las que se transmiten asíncronamente mensajes, permitiendo que aunque un vehículo no pueda contactar con otro para que el mensaje llegue al destino, pueda esperar un tiempo y al desplazarse el vehículo cambiar de zona donde sí se encuentren más vehículos a los que transmitir la información. Este paradigma se denomina almacenamiento, transporte y reenvío.

En el caso de las redes DTN la alta movilidad de los vehículos se convierte en una oportunidad, ya que en entornos grandes en los que haya que recorrer largas distancias es muy difícil crear una ruta en un instante determinado de tiempo que consiga ir desde el origen al destino saltando de un vehículo a otro, pues habrá zonas con una densidad de tráfico muy baja donde no se podrán encontrar vehículos. Partiendo la comunicación en saltos que pueden ser distantes en el tiempo es más sencillo que la información viaje a través de la red.

La transmisión de información entre dos vehículos se puede realizar de forma totalmente casual, en la cual un vehículo encuentra a otro que está más cerca del destino y le envía la información para que él se encargue, puede estar planificada, ya que si se conoce la posición de los vehículos en el mapa y la dirección en la que se mueven, se puede planificar el momento en el que dos vehículos pueden entrar en contacto. También es posible predecir el contacto con otro vehículo analizando el historial de contactos que se han realizado con él anteriormente, y así prediciendo en qué condiciones se producirá el siguiente.

Por último, existen dos modelos de realización de la transmisión en DTN. El primero está basado en el reenvío a vehículos seleccionados con el objetivo de que ellos se encarguen de alcanzar el destino. El otro modelo se basa en la inundación, en el cual se envía la información a todos los vehículos cercanos, estos a los suyos, y así sucesivamente hasta que alcance el destino. Este modelo supone una mayor sobrecarga de la red, ya que malgasta los recursos de la misma, y conlleva una disminución de prestaciones.

2.2.3. VDTN

El tercer tipo de red vehicular (y también el más reciente) es una DTN en la cual se aplican principios tomados de la conmutación óptica, en particular la conmutación a ráfagas.

La principal diferencia de las redes VDTN y las DTN es la separación que se hace en la primera de ellas entre el plano de control y el plano de datos. Los vehículos disponen de un ancho de banda limitado, ya que en aglomeraciones muy grandes de vehículos si todos consumen mucho ancho de banda, la red no lo soportaría por el solape espectral que acarrea, dando lugar a numerosas colisiones y a que sea muy difícil transmitir. Es por eso que se añade en las redes VDTN un plano de control que, consumiendo muy poco ancho de banda, permite que más vehículos puedan transmitir simultáneamente, por medio de limitar el uso del plano de datos al momento en que la comunicación tiene una alta probabilidad de éxito. Así se evita malgastar ancho de banda en el envío de datos cuando no van a llegar a su destino.

Sin la inclusión de un plano de control, los vehículos pueden transmitir varias veces la información sin éxito pues quizá no se encuentran vehículos cerca, o no están en disposición de recibir. En cambio, gracias al plano de control se negocia el envío previamente, y cuando éste se realice se hará con éxito, evitando el gasto innecesario de recursos.

En las redes VDTN se consideran dos tipos de nodos, los cuales pueden representar tres papeles distintos. El primer tipo son los vehículos, que pueden hacer el papel de nodo terminal, que son los que inician la comunicación o son el destinatario de la misma. Un ejemplo de este tipo puede ser un vehículo que solicite descargar información de la situación actual del tráfico. El otro papel que puede realizar un vehículo es el de nodo móvil, los cuales se encargan de transportar la información, obteniéndola de vecinos y reenviándola cuando se hayan desplazado para servirla en zonas donde esta información no se encuentre. El segundo tipo de nodo es el repetidor (que corresponde con lo que hemos denominado estaciones de información en el proyecto), los cuales se encuentran en las intersecciones más importantes donde se aglomera un mayor número de vehículos y almacenan toda la información que pueden para servirla a otros vehículos que puedan estar interesados en la misma. Así, siguiendo el ejemplo del nodo terminal interesado en descargar información del tráfico de la zona en la que está circulando, el vehículo terminal podría descargarse esta información al pasar por un nodo repetidor, o podría realizar esta acción un nodo móvil, e ir pasándola de uno a otro hasta que alcanzara al nodo terminal.

Para el desarrollo de este trabajo, el tipo de red vehicular utilizado será una DTN, debido a que la naturaleza de los contenidos de información y entretenimiento no requiere que los datos se sirvan en el instante en el que se requieren, sino que hay un cierto tiempo de retardo aceptable, además en el sistema que se desarrolla en el proyecto no se genera mucha carga en la red, por lo que no es esencial cuidar la cantidad de espectro que usa cada vehículo (inserción de plano de control). Aun así, algunos de los protocolos utilizados utilizan mensajes de señalización para evitar el envío de contenidos que no llegarán a su destino.

2.3. Aplicaciones

Para comprender el interés que puede suscitar el dotar de capacidad de conexión a la red de vehículos que hasta ahora no la poseían o la poseían de forma muy limitada, es necesario mencionar la cantidad de aplicaciones a las que se pueden dar lugar.

Como se enuncia en [4], las aplicaciones que pueden desembocar tras la implementación de redes tipo VANET se dividen en tres grandes grupos: Aplicaciones relacionadas con la seguridad vial, aplicaciones relacionadas con la eficiencia y gestión del tráfico y aplicaciones relacionadas con la información y el entretenimiento. A continuación se pasará a hablar de cada una de ellas.

2.3.1. Seguridad vial

En un primer momento, la idea de conectar los vehículos a la red vino inspirada por la necesidad de mejorar la seguridad en las carreteras, ya que puede suponer que se salven centenares de vidas al año en un único país, o que los daños puedan ser paliados en gran medida.

Dado que una buena parte de los accidentes que se producen anualmente están producidos en intersecciones y colisiones frontales, traseras o laterales, el hecho de que se pueda tener

información en tiempo real de los vehículos que cada uno tiene a su alrededor como la posición, la velocidad o la dirección, se puede explotar este conocimiento para evitar tales accidentes, lo que supondría una disminución muy notable en el número de siniestros producidos en las carreteras. Además, se podrá actualizar a través de la compartición de información entre vehículos situaciones anómalas de la carretera, como puede ser desprendimiento de rocas, formación de placas de hielo o generación de baches en la carretera, lo que puede alertar a los conductores en un intervalo muy corto de tiempo sin que tengan que intervenir las autoridades para señalar los obstáculos.

A continuación se enumerarán algunas de las principales aplicaciones relacionadas con la seguridad vial:

Aviso de colisión en una intersección: El aviso de la proximidad de un vehículo cuando se desea realizar un cruce puede alertar a tiempo a uno de los dos conductores para que frenen a tiempo y no se produzca una colisión.

Asistencia en el cambio de carril: Aviso de peligro en caso de que se esté realizando un cambio de carril y un vehículo se encuentre en el ángulo muerto donde no hay visibilidad.

Asistencia en adelantamientos: Realización de avisos cuando dos vehículos intentan realizar simultáneamente un adelantamiento sin que ambos se den cuenta.

Aviso de colisión frontal: Notificación temprana a dos vehículos que estén avanzando en el mismo carril en direcciones opuestas, evitando que alcancen una distancia mínima en la que la reacción no puede prevenir la colisión.

Aviso de colisión trasera: En caso de que no se respete la distancia de seguridad, sobre todo en puntos con alto riesgo de congestión de tráfico o cambios de pendiente que pueden ocasionar frenazos bruscos, se le notificará al vehículo que tiene que aumentar la distancia con el vehículo que tiene al frente.

Aviso de riesgo de accidente: En caso de que se detecte un riesgo alto de accidente por las condiciones del tráfico actuales, los vehículos cooperarán entre ellos para reorganizarse y evitar que el accidente se produzca.

Aviso de vehículo de emergencias: Cuando un vehículo de emergencias como una ambulancia o un vehículo policial necesiten abrirse camino para avanzar con más velocidad, se le informará a los vehículos que deben dejar un carril libre.

Actuación ante colisión inminente: Dado que se monitoriza continuamente el tráfico, en caso de que una colisión ya sea inminente y no pueda evitarse, se tomarán las medidas necesarias para paliar el daño, como lanzamiento anticipado de los airbags o cinturones de seguridad motorizados que puedan tensarse antes para evitar movimientos más bruscos.

Asistencia en cuellos de botella: En puntos donde se encuentre una fusión de varios carriles en uno, los vehículos negociarán entre ellos para facilitar el momento en el que cada uno se incorpora.

Frenado de emergencia: En momentos en los que un vehículo tenga que realizar un frenado de emergencia, se avisará inmediatamente al resto de la situación para que el tiempo de reacción sea menor.

Aviso de circulación en sentido contrario: Si un vehículo está circulando en un sentido no permitido se avisará al resto de vehículos de esta situación para que estén alerta.

Aviso de vehículo estacionado: Si un vehículo necesita detenerse por una emergencia, se avisará al resto de su situación.

Aviso de condiciones de tráfico: Cuando un vehículo detecta un cambio en las condiciones del tráfico (congestión, aglomeraciones), lo comunica inmediatamente al resto.

Riesgo de colisión: En el caso de que se detecte que dos vehículos tienen un riesgo alto de colisión y estos dos vehículos no pueden comunicarse, se compartirá a todos los vehículos de la zona dicha información.

2.3.2. Eficiencia y gestión del tráfico

Las aplicaciones relacionadas con la eficiencia y la gestión del tráfico se ocupan de mejorar la fluidez del tráfico en aras de evitar atascos o saturación de una vía existiendo otras en las que apenas hay flujo de vehículos. Esta actividad se realiza a través de la asistencia a vehículos aportándoles información útil, como pueden ser mapas o eventos significativos como un accidente o un corte de carretera que esté desembocando en atasco. Dentro de este grupo, se suele hacer la división en gestión de velocidad y navegación cooperativa:

Gestión de velocidad: Las aplicaciones relacionadas con la gestión de la velocidad tienen la función de asesorar al conductor sobre la velocidad ideal a la que debe circular para evitar paradas innecesarias que pueden conllevar un gasto de tiempo y de dinero mayor.

Navegación cooperativa: La navegación cooperativa supone la organización de los vehículos de la zona para establecer un modo de actuación común óptimo basado en la cooperación mutua para que el tráfico en la zona sea fluido y no haya factores que retrasen la circulación.

2.3.3. Aplicaciones de información y entretenimiento

A pesar de que los dos grupos anteriormente mencionados son los que primero se idearon para el uso de redes vehiculares, el tercer grupo es el que requerirá más recursos de la red al poder ser utilizado bajo demanda del usuario, las aplicaciones de información y

entretenimiento. Este grupo será el que se utilizará como referencia para el desarrollo del proyecto.

Para hablar de aplicaciones de este tipo, en [5] y [6] se establecen dos subgrupos relacionados con el alcance de los contenidos: Contenidos de carácter local y contenidos de carácter global.

Contenidos de carácter local: Las aplicaciones pertenecientes a este subgrupo tienen una utilidad basada en la localización del usuario, es decir, el contenido puede ser atractivo para una persona o no en función del área geográfica donde esté. Por ejemplo, en caso de que un usuario reciba una notificación de que una gasolinera ha bajado un 10% el precio del litro de gasolina, esta información será de interés si el vehículo está cerca de esa gasolinera, y no a 300 km. En este grupo podemos ubicar aplicaciones como identificación automática para acceso a establecimientos y aparcamientos, notificaciones de ubicación cercana de puntos de interés del usuario o cobro electrónico de peajes.

Contenidos de carácter global: Este tipo de aplicaciones se refieren a las que comúnmente se pueden encontrar en Internet o simplemente servicios que no requieren una limitación geográfica, sino que son válidos en cualquier localización. La descarga de contenidos audiovisuales como por ejemplo música o películas serían contenidos dedicados al entretenimiento que pertenecerían a este grupo. Por otro lado, información necesaria para el mantenimiento del sistema como actualización de datos o calibración también se incluiría dentro de este grupo.

Como se ha podido observar, las posibilidades que ofrece la implantación de este tipo de redes son muy variadas, además de poder ser útiles para la movilidad en carretera, previniendo accidentes u optimizando la fluidez del tráfico.

2.4. Regulación actual

Como en cualquier proyecto tecnológico que se desee llevar a cabo, es necesario un proceso de estandarización de tecnologías, establecer qué protocolos están permitidos, qué rango de frecuencias podrán utilizar, compatibilidad entre protocolos, etc. En esta sección se desarrollará una visión general del proceso de creación de estándares y proyectos que se han puesto en marcha en las tres regiones más significativas: Estados Unidos, Japón y la Unión Europea. Evidentemente, una tecnología como la que se trata en este trabajo, la cual todavía no ha sido puesta en marcha de forma masiva, será muy heterogénea en función del área geográfica en la que se esté desarrollando, y cada región estará tomando diferentes formas de abordar el problema. La recopilación completa de la cual se ha extraído la información se puede encontrar en [4].

2.4.1. Estandarización en Estados Unidos

El origen de esta tecnología en Estados Unidos se remonta a 1991, cuando en el Congreso se propuso la creación de un organismo dedicado a mejorar la eficiencia y la seguridad en las carreteras norteamericanas mientras que se reducían los niveles de contaminación

y consumo de combustibles fósiles. Este programa fue llamado IHVS (Intelligent Vehicle Highway Systems). A partir de entonces comenzó a utilizarse el concepto de ITS (Intelligent Transportation System) para referirse a las tecnologías relacionadas con las redes vehiculares.

Para el uso del espectro electromagnético es necesario establecer una banda de frecuencia dedicada a la tecnología que se quiere poner en marcha, y fue en 1999 cuando se reservó una banda de 75 MHz en los 5.85-5.925 GHz. El siguiente paso fue establecer cuál sería el estándar utilizado para la transmisión en el canal. Se decidió que debería estar basado en uno de los organismos de estandarización internacional más relevantes del sector de la ingeniería como es el IEEE (Institute of Electric and Electronic Engineers), concretamente el estándar IEEE 802.11, que es el encargado de las redes de área local inalámbricas. Es por eso que a partir de 2004 se comenzó a desarrollar un estándar específico sobre comunicaciones inalámbricas en entornos vehiculares que actualmente se conoce como IEEE 802.11p [7].

El estándar anteriormente comentado se refiere solamente a la capa física y a la capa de enlace según el sistema OSI [8], por lo que el grupo 1609 del IEEE comenzó a trabajar en estándares del resto de capas, dando lugar a los estándares IEEE 1609.1 (relacionado con el control de recursos), el IEEE 1609.2 (relacionado con la seguridad), IEEE 1609.3 (relacionado con la red) y el IEEE 1609.4 (operación multicanal). El compendio del 802.11p y el IEEE 1609 dieron lugar al protocolo WAVE (Wireless Access in Vehicular Environments), el cual es el utilizado en la mayoría de proyectos que se han puesto en marcha en Estados Unidos.

Por otro lado, otra organización relacionada con la estandarización a nivel norteamericano como la SAE (Society of Automotive Engineers) ha estado colaborando activamente con el grupo IEEE 1609 anteriormente comentado para la creación de un estándar de mensajes que permitan la comunicación a través de los protocolos que propuso el grupo del IEEE.

2.4.2. Estandarización en Japón

Es necesario hacer mención al proceso de estandarización en el país nipón debido a la alta inversión que ha realizado en el desarrollo de tecnologías ITS y redes vehiculares. En 1996 se presentó el “Plan integral para las ITS en Japón”, el cual contó con el respaldo de 5 figuras importantes en el gobierno. El propósito que perseguía este plan era el establecimiento de las líneas básicas para el avance en la sociedad de la Información y Telecomunicación. En el año 1999, el diseño de la arquitectura de los ITS se había realizado.

Un dato interesante, y que es representativo del nivel de desarrollo en este campo que existe en Japón, es que en 2008 aproximadamente 20 millones de vehículos ya estaban en posesión de tecnología de comunicación entre vehículos.

2.4.3. Estandarización en la Unión Europea

Por último, se tratará el proceso de estandarización de las tecnologías ITS en el conjunto de la Unión Europea, la cual tiene como objetivo beneficiar a todos los conductores de las carreteras europeas en temas de seguridad vial y mejora del tráfico. El hecho de que existan organismos que velen por el desarrollo y la estandarización de la tecnología a nivel europeo resulta muy beneficioso para el sector ya que el sistema tiende a ser más homogéneo que en el caso de que cada país aplicara sus propios criterios de regulación.

Las instituciones europeas encargadas de los estándares a nivel europeo son el CEN (European Committee for Standardization), el CENELEC (European Committee for Electro-technical Standardization) y el ETSI (European Telecommunications Standards Institute).

El CEN es el organismo que actualmente ha estandarizado la franja de frecuencia DSRC (Dedicated Short-Range Communications) en 5.9 GHz. Por otro lado el ETSI ha creado distintos grupos de trabajo: WG1, el cual describe los requisitos de la aplicación, WG2 que trabaja en la arquitectura, WG3 que se encarga del protocolo de red y transporte, WG4 es el encargado del desarrollo del perfil europeo del 802.11p, y WG5 que se encarga de la seguridad.

Las instituciones europeas anteriormente mencionadas trabajan estrechamente con otras organizaciones de estandarización internacionales como el ISO (International Organization for Standardization), el IEC (International Electro-technical Commission) y la ITU (International Telecommunication Union).

La ISO creó en 1993 el estándar ISO/TC 204 que está relacionado con las actividades ITS. El protocolo que se ha desarrollado desde este organismo es el llamado CALM (Continuous Air-interface Long and Medium range), el cual considera distintas tecnologías como comunicaciones infrarrojas, GSM o CEN DSRC.

La organización ERTICO ITS es una iniciativa de la Comisión Europea y los ministros de Transporte e Industria, la cual aglomera a todo el sector de las ITS en Europa. La función principal de este organismo es la investigación y el desarrollo de las ITS.

El C2C-CC (Car 2 Car Communication Consortium) es también una organización de carácter europeo creada por los fabricantes de automóviles que trabaja junto con el ETSI y el ISO para la creación de nuevos estándares.

2.5. Entorno socio-económico

Durante los últimos años, la economía mundial ha estado marcada por un periodo de recesión muy importante que ha afectado a todos los sectores, y ha producido en consecuencia una importante bajada del consumo por la pérdida de poder adquisitivo de los ciudadanos. Estos años en los que tanto los estados como las empresas han sido más

conservadores con el gasto que realizaban, la investigación ha sido uno de los más castigados y se ha ralentizado en gran medida.

En cambio, en algunos países como España o Estados Unidos se ha observado un pequeño repunte de la economía en este último año, y la venta de vehículos que era uno de los sectores más castigados se ha recuperado notablemente, lo que supone una oportunidad para el sector del automóvil de implementar nueva tecnología que resulte más atractiva para los compradores potenciales, tecnología como la que se trata en este proyecto.

El factor económico que resulta más influyente en España es la tasa de paro, una de las más altas de Europa. En una población en la que más de un cuarto de la misma no tiene posibilidad de acceder a un empleo no se permitirán gastos que no cubran necesidades básicas, y si necesitan comprar un vehículo porque es esencial para el desarrollo de su vida cotidiana será de un precio lo más bajo posible, por lo que la inclusión de esta nueva tecnología no puede permitirse elevar en gran medida el precio de los vehículos, pues este tipo de productos tienen una demanda muy elástica.

En cuanto a los factores sociales, las personas cada vez están más dispuestas a utilizar la última tecnología y se muestran menos reticentes a adquirir nuevos productos. Las generaciones jóvenes que tienen edad para conducir han crecido rodeados de tecnología y se han formado en la era digital, por lo tanto tienen una mayor predisposición a estar interesados en nuevas fuentes de acceso a contenidos digitales. En cambio, en conductores de una edad más avanzada la inserción de esta funcionalidad puede resultar difícil de utilizar, aunque sin duda se puede potenciar la utilidad de las aplicaciones de gestión de tráfico, que permitirán ahorrar tiempo en atascos.

Por otro lado, la población de los países occidentales tiende al éxodo rural, lo que cada vez produce que los núcleos de población estén más concentrados, y consecuentemente que el tráfico de vehículos sea mayor. Esto supone una oportunidad perfecta para el uso de esta tecnología, pues su fin es mejorar la calidad de vida de las personas cuando éstas están dentro de un vehículo.

Por último, las normativas y políticas públicas se están homogeneizando y avanzan a una mayor velocidad gracias a la Unión Europea, la cual permite regular con un mismo criterio varios países y agrupar intereses comunes como se ha visto en el apartado de regulación, en el cual varios organismos cooperan para facilitar el avance y la inversión en nueva tecnología.

2.6. Protocolos de distribución de contenidos en VANETs

Como se ha indicado en apartados anteriores, para la realización del proyecto se centrará el análisis en la distribución de contenidos dentro de las redes vehiculares, por lo que no se tendrán en cuenta las aplicaciones relacionadas con la seguridad vial, sino aquellas en las que un vehículo desee obtener un contenido de cualquier tipo, ya sea información o entretenimiento. Este tipo de aplicaciones no requieren la obtención inmediata del

contenido, ya que los usuarios suelen tolerar esperar unos segundos para ver el estado del tráfico, o cuáles son las gasolineras con los precios más bajos, pues al estar conduciendo esta tarea no es crítica.

Los entes encargados de proveer de contenidos a los vehículos son las estaciones de información, las cuales se encontrarán fijas en los puntos con más confluencia de tráfico de la ciudad, aunque si un vehículo ya ha almacenado un contenido puede proveer a otro que quizá no lo tenga y lo esté buscando. Este modelo coincide con el presentado en las redes DTN, en las que los vehículos almacenan, transportan el contenido a través de distintas zonas y lo reenvían, lo que permite una gran difusión.

Los protocolos de distribución de contenidos se pueden dividir en dos grandes familias [9], los protocolos de tipo *Push* (Cuyo significado en inglés es empujar), y los protocolos de tipo *Pull* (Tirar).

El principio de funcionamiento de los protocolos del primer tipo es que un vehículo o una estación de información envían cada cierto tiempo un contenido a sus vecinos en *broadcast*, y así un vehículo que pueda estar interesado en dicho contenido puede utilizarlo. De ahí que el nombre del protocolo signifique “empujar”, pues van disseminando sus contenidos a través de la red (Ilustración 2-2).

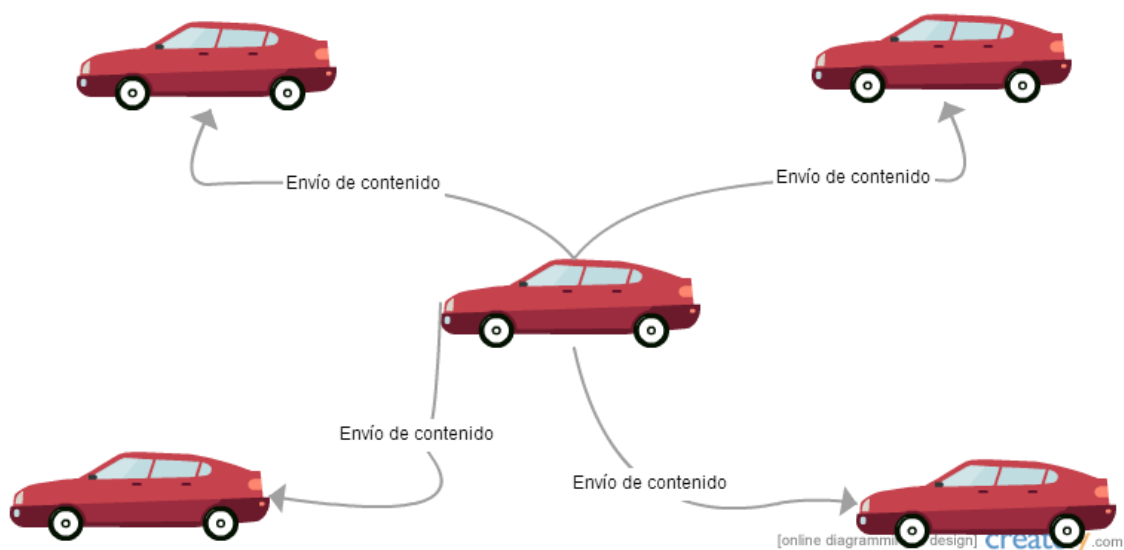


Ilustración 2-2. Funcionamiento de los protocolos tipo Push

En cambio, los protocolos de tipo *Pull* son más selectivos a la hora de enviar un contenido, y sólo lo sirven bajo petición previa, es decir, cuando un vehículo necesita obtener un contenido, el vehículo realiza una petición a todos sus vecinos (ya sean vehículos o estaciones de información) para indicarles que si alguno tiene dicho contenido tiene que enviárselo. Esto explica el nombre de esta familia de protocolos, pues tienen que “tirar” del contenido para obtenerlo (Ilustración 2-3).

Como se puede deducir, los protocolos de tipo *Push* introducirán más sobrecarga a la red, pues estarán continuamente enviando contenidos aunque ninguno de sus vecinos esté

interesado en el mismo, pero en cambio realizan una mayor difusión de contenidos por la red, pues cada vehículo se encargará de hacer *broadcast* de los mismos cuando se vaya moviendo por las distintas carreteras de una ciudad, produciendo que los vehículos tengan almacenados en su memoria una mayor cantidad de contenidos.

En cambio, los protocolos de tipo *Pull* no sobrecargan tanto la red pues sólo transmiten un contenido cuando otro lo solicita y el mensaje de petición es mucho menor que un mensaje que contenga un contenido, aunque esto supone una desventaja a la hora de diseminar un contenido en varias zonas geográficas, pues hasta que un vehículo cercano no solicite dicho contenido, este no será enviado, y los vehículos almacenarán en su memoria sólo el contenido que a ellos les interesa.

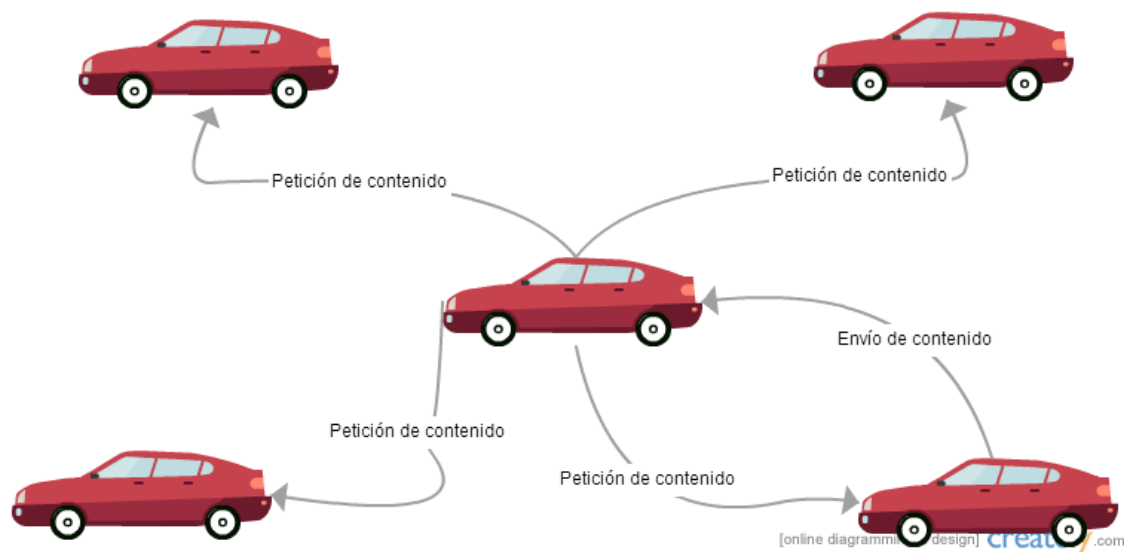


Ilustración 2-3. Funcionamiento de los protocolos tipo Pull

A continuación, se hará un estudio de diferentes propuestas de protocolos de distribución de contenidos que se han realizado en los últimos años, mencionando las principales características de los mismos. En primer lugar, se tratarán protocolos de la familia de los *Push*, puesto que un gran número de protocolos implementan este sistema.

En primer lugar, en el protocolo propuesto en [10] se considera un escenario en el que los vehículos se mueven en línea recta a través de un escenario sin obstáculos, y se realizan cambios en el radio de alcance de los vehículos, en la velocidad y en la densidad de tráfico. El número de contenidos en la red es igual al número de vehículos, y no existe popularidad, así que cada contenido es igualmente probable. Además, se propone una función de relevancia en la cual los vehículos eligen qué contenidos almacenan y cuáles no (pues la memoria no es ilimitada). La descarga de contenidos utilizando una red 3G o 4G no se contempla en este caso de estudio.

En las simulaciones realizadas en [11] también se considera un número de contenidos igual al número de vehículos, sin popularidad ni posibilidad de descargar contenidos usando la red celular, aunque en este protocolo el temporizador no es fijo, sino adaptativo a la hora de enviar el contenido a sus vecinos, lo que produce que en situaciones con poca

movilidad los vehículos no transmitan con tanta frecuencia pues sus vecinos ya habrán recibido los contenidos almacenados. En este caso los vehículos sí se mueven utilizando carreteras, considerando tráfico lento y tráfico normal, variando la velocidad y la densidad de tráfico.

En el caso de [12], el número de contenidos que se han considerado varía entre 10 y 600, sin popularidad ni 3G. Se realizan las simulaciones utilizando un mapa real, concretamente la ciudad de State College, en el estado norteamericano de Pensilvania. El número de vehículos utilizado en la calle principal ha sido de 150, y en las calles secundarias de 60. Lo más novedoso que incluye este trabajo con respecto a los dos anteriores es la inclusión de estaciones de información que almacenan contenidos y los reenvían, estando situadas en las zonas de intersección de más vehículos.

Las propuestas anteriores mostraban la aplicación de protocolos tipo *Push* cambiando los intervalos de tiempo en los que se reenvía o los contenidos que almacenan. En [13] se añade el criterio de la posición de los vehículos para el reenvío de contenidos. El funcionamiento es el siguiente: La estación de información envía a sus nodos vecinos un contenido, y elige a dos de los que lo reciben para que estos lo reenvíen, y a su vez estos podrán elegir los nodos, por lo que se puede planificar las zonas en las que se irá diseminando el mensaje, además de que las estaciones especifican el tiempo de vida del mismo, por lo que cuando este haya pasado los vehículos dejarán de reenviarlo. El problema de esta propuesta es que solamente se considera un contenido en la red, lo que resulta poco realista. Tampoco se tiene en cuenta conectividad a la red celular.

El concepto de la inclusión del *geocast* en el sistema también se considera en [14] a través de 3 propuestas distintas: En la primera, un servidor central será el encargado de diseñar la distribución que se hará de contenidos. En la segunda, un nodo será el elegido para almacenar el contenido y distribuirlo cada cierto tiempo, o cuando se le notifique. En la tercera, todos los nodos almacenan los vecinos que hay en la zona, y cuando se detecta un nuevo nodo se le envía el contenido. Como en [13], también se considera únicamente un contenido a la hora de realizar las simulaciones sin opción de utilizar conexión a una red 3G, además de que en esta propuesta el protocolo propuesto no es puramente *Push*, sino que es híbrido con los protocolos del tipo *Pull*.

En [15] y [16] se proponen mecanismos de distribución utilizando *Push* y criterios basados en la localización añadiendo varios contenidos, lo cual hace más realistas las simulaciones y sus resultados pueden ser interpretados con mayor confianza, aunque en estos casos tampoco se contempla popularidad ni posibilidad de descarga de contenidos por 3G.

Por otro lado, también se han publicado muchos trabajos en los que se considera más apropiado el uso de *Pull* para distribuir contenidos a través de la red, en los cuales es necesario recordar que la transmisión se realiza bajo demanda.

En [17] se ha utilizado un modelo de *Pull* en el cual las peticiones solo se realizan a los vecinos a los que les llega la petición (que en escenario son todos los que estén en un radio de 250 metros), y estos le contestan si poseen el contenido, o en el caso de que sea

una estación la que está cerca, le envía siempre el contenido, pues las estaciones están conectadas a Internet. Se han considerado para las simulaciones distintos escenarios con diferentes densidades de tráfico y velocidades (escenarios urbanos, rurales, autovías, etc.), mientras que el número de contenidos utilizado ha sido de 1, lo que como se ha comentado en protocolos anteriores hace que los resultados no se aproximen a casos reales, además que la probabilidad de obtener un contenido tras realizar la petición aumenta notablemente cuando los vecinos están interesados en el mismo, pues lo conseguirán tarde o temprano.

Por otro lado, en [18] cambia el modelo de peticiones y ya no se realiza a los vecinos que se encuentran a un único salto, sino que ésta llega a los vecinos que se encuentran a una mayor distancia. Además en este trabajo se incluyen varios contenidos (de 4 a 32) y se considera la opción de que exista popularidad (con un parámetro alfa de 0.8 en una distribución Zipf, lo que indica un nivel de popularidad media), y los contenidos están divididos en trozos para considerar el envío fraccionado.

En todas las propuestas que se han mencionado hasta el momento, en ninguna se tiene en cuenta la posibilidad de añadir la opción de descarga por 3G si el contenido no se puede obtener a través de la VANET, una opción que es necesaria ya que a pesar de que los usuarios pueden tolerar cierto retardo, lo que no es válido es que no reciban el contenido en ningún momento, o 30 minutos después, porque en ese momento el contenido ya no será útil. En [19] se utiliza la red 3G para descargar información de control como es la conectividad en distintas zonas, para que los vehículos puedan tener la capacidad de saber si en una zona van a poder encontrar un contenido o no va a ser posible, además que esta información se puede utilizar para el proceso de encaminamiento al poder elegir las rutas donde hay más conectividad. En [20] se usa la red 4G para poder formar distintos grupos de vehículos y favorecer la comunicación.

Para solventar las bajas prestaciones de las VANET cuando el número de contenidos es elevado, en [9] se propone un método *Pull* multi-salto cuyo algoritmo está basado en el *routing* geográfico. Este algoritmo se basa en la transmisión a un destinatario en función de su posición geográfica, pues buscará un vehículo que lo acerque más a su destino. El destino ideal para conseguir un contenido es una estación de información, ya que puede proporcionar acceso a cualquier contenido, y es por eso que éste será el destino de todas las transmisiones que se realizan siguiendo este algoritmo.

Para que la petición llegue a una estación, el vehículo emisor posee las coordenadas de todas las estaciones de información, y selecciona a la más cercana, para elegir como nodo que reenviará su petición a aquel cuya posición esté a una distancia euclídea menor, y así se irán repitiendo las peticiones hasta llegar a la estación de información, o en el caso de que alguno de los nodos realice una petición y algún nodo vecino posea el contenido, el algoritmo converge y el contenido se envía en el orden inverso en el que se han realizado las peticiones hasta que el contenido llega al nodo emisor de la petición. En el caso de que en alguno de los saltos la petición no sea escuchada por ningún vehículo que se

encuentre más cerca de la estación de información, el algoritmo finaliza y el nodo emisor debe volver a realizar la petición al transcurrir un tiempo establecido.

Cambiando de método de aplicación, en [21] las comunicaciones no se realizan mediante una red DTN, y el objetivo del sistema no es la distribución de contenidos sino la comunicación entre vehículos. La red celular se utiliza para poder comunicarse con el vehículo deseado, pero también se comprueba si hay posibilidad de comunicarse con ese vehículo a través de la VANET, y en caso de ser posible, se realiza la comunicación por esta vía, aliviando así la carga de la red celular. Este paradigma cambia en [22], en el cual se estudia la división de flujos de tráfico para la red celular o para la VANET en función de la naturaleza de estos flujos. En caso de que los flujos requieran características que la VANET no pueda garantizar, como por ejemplo las aplicaciones de voz IP que requieren retardos mínimos, éstas se realizan por la 3G, mientras que aplicaciones como la navegación web sí se realizan por la red vehicular.

Una vez revisada una muestra de la literatura generada en torno a la distribución de contenidos en VANET, es necesario hacer un análisis general. En la mayoría de artículos revisados, el número de contenidos en la red era muy bajo (siendo 1 en muchos casos), y en los que no era así se ha podido demostrar que las prestaciones de la red se ven muy afectadas por el aumento del número de contenidos. La explicación de este fenómeno es muy lógica, ya que si el número de contenidos es muy grande, la probabilidad de que un vehículo que utiliza un protocolo tipo *Push* haga *broadcast* de uno de los contenidos que tenga almacenados y que uno de los vecinos esté interesado en el mismo es muy baja. En los protocolos de tipo *Pull* la situación no mejora sustancialmente, pues los vehículos pedirán un contenido concreto y será muy difícil que precisamente uno de sus vecinos sea el que lo tenga.

A pesar de que no estén en funcionamiento redes vehiculares que permitan a los usuarios descargar contenidos, sí se puede presuponer que el modelo que seguirá será muy parecido a un modelo como el de Internet, en el cual los usuarios solicitan una gran variedad de contenidos. Aunque es cierto que en las redes vehiculares se puede explotar el hecho de que se busque información centrada en la localización del usuario, como puede ser búsqueda de lugares cercanos de interés, y que otros vehículos de la zona ya habrán utilizado y tendrán almacenados, existirán contenidos de carácter global que serán menos solicitados por los vehículos de una zona.

Esto hace alusión al concepto de la popularidad de los contenidos, que en algunos de los últimos protocolos revisados aparecía. La consideración de que unos contenidos sean más populares que otros, y que por lo tanto sea más probable que los vehículos los pidan en la red, mejora ligeramente el problema de la existencia de varios contenidos, pero con niveles de popularidad no muy elevada tampoco debe ser una solución milagrosa.

En algunos protocolos de tipo *Pull* se ha contemplado la posibilidad de realizar varios saltos para llegar a un nodo que sí pueda tener opción de servir el contenido porque lo tenga almacenado (ya sea un vehículo o una estación). Este planteamiento sugiere que el rendimiento de la red será mucho mejor en este caso, pero sin saber qué impacto tendrá

realmente en la sobrecarga de la red o en la tasa de éxito con la existencia de varios contenidos.

Para solventar estas cuestiones, se propone en este trabajo un desarrollo similar al que se ha llevado a cabo en [9], considerando protocolos de tipo *Push* y de tipo *Pull* a un salto o a varios (a varios saltos se propone en dicho trabajo un protocolo propio llamado MHTI). Se estudiará el impacto que tiene la densidad de tráfico, la existencia de una gran cantidad de contenidos o de popularidad, pero el software utilizado en este caso será distinto, siendo más rápido en la ejecución al no tener en cuenta modelos de transmisión físicos como la atenuación o las colisiones en la red.

2.7. Simulación basada en agentes

La simulación basada en agentes representa una forma de modelización de sistemas complejos que están compuestos por un número elevado de objetos activos como pueden ser por ejemplo personas, vehículos o animales, los cuales son llamados agentes dado que tienen capacidad de ser autónomos. Cada agente está “programado” para que tenga un comportamiento determinado, el cual se puede ver influido por factores que no dependan de él.

Como se enuncia en [23], los modelos basados en agentes autónomos comparten una estructura común: Todos tienen un grupo de agentes con unos atributos que representan sus características propias y un comportamiento determinado, ambos definidos desde el primer momento en que entran en la simulación. Por otro lado, hay definidos métodos de interacción y comunicación entre los agentes. Por último, todos los agentes se encuentran en un escenario específico con el cual también pueden interaccionar, y también les condicionará su comportamiento.



Ilustración 2-4. Atributos y métodos de un agente vehículo

La característica principal de un agente es la autonomía, pero existen otras características que también lo definen como es el hecho de estar auto contenido, es decir, tiene una identidad propia independientemente del resto, y se puede distinguir cada uno de sus atributos. Otra característica es que los agentes tienen un estado, en el caso de un vehículo en una VANET este estado puede consistir en buscar un contenido, o el estado inicial en el que todavía no está interesado en ninguno. También es característico el hecho de que puedan tener una finalidad, como puede ser en este caso la obtención misma de contenidos. Todas estas características se pueden visualizar en la Ilustración 2-4 la cual representa los atributos y métodos (acciones que puede realizar) un vehículo que actúe como un agente en una simulación de una red vehicular. La identidad se la otorga el identificador, que en este caso es el 10 (único para cada agente), y actualmente se encuentra en estado de búsqueda de un contenido, concretamente el contenido con identificador 8. Durante el tiempo que ha estado en la simulación ha obtenido 3 contenidos que no necesitaba en ese momento, pero que puede compartir con otros agentes a través de su método enviar contenido. Otra de las acciones que puede realizar es moverse a través de la ciudad, lo que desencadenará en que actualice su posición, que en el momento concreto en que se ha observado es de 135 en el eje horizontal y 250 en el eje vertical.

En cuanto a las interacciones que pueden tener los agentes entre ellos, suelen estar delimitadas a un grupo específico como suelen ser los agentes que se encuentran más próximos a ellos. Esto es debido a que los agentes solo tienen información local, no información de lo que ocurre en toda la red que se está simulando. Siguiendo con la analogía de una simulación de una red vehicular, como se puede observar en la Ilustración

2-5 a la izquierda se encuentra el vehículo que en el párrafo anterior se ha analizado, el cual interacciona con el vehículo 4, el cual le envía el contenido que este estaba buscando, por lo que su estado cambia a obtenido conseguido, y el contenido 8 pasa a quedar almacenado.

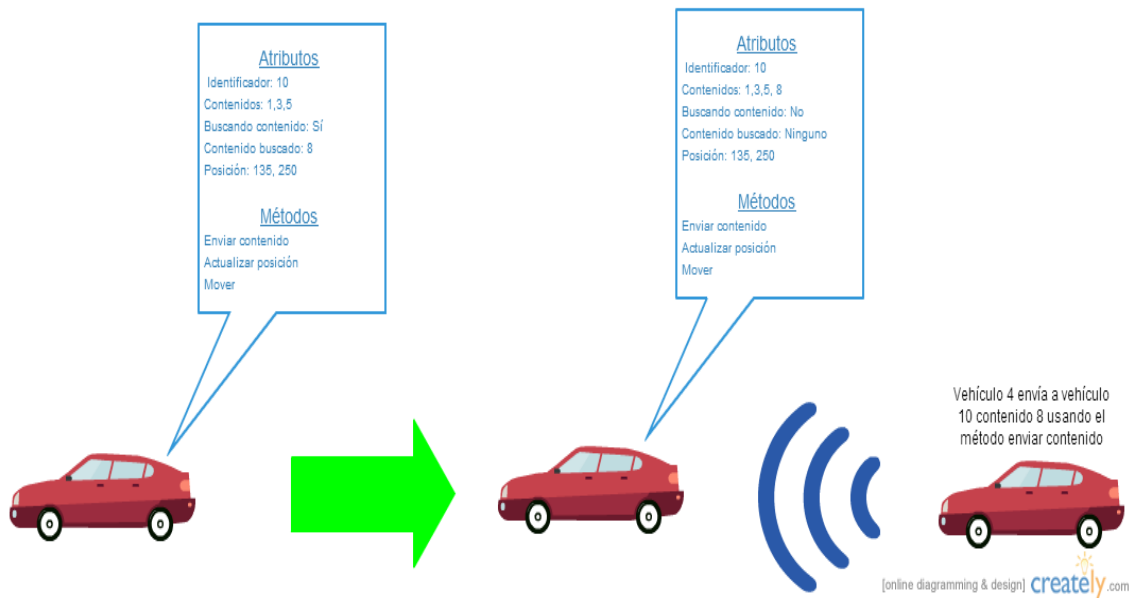


Ilustración 2-5. Comunicación entre agentes a través de sus métodos y cambio de sus atributos.

El medio que circunda al agente también es determinante para definir tanto sus atributos como sus acciones, ya que dependiendo en la zona en la que se encuentre podrá moverse a una velocidad o a otra o tendrá definidas unas características específicas para ese entorno. Volviendo a utilizar la comparación con una simulación de una red que interconecte vehículos, ya que esta va a ser la que se va a realizar en el proyecto, en la Ilustración 2-6 se puede ver lo que condiciona el medio a la hora de interactuar los agentes. Dos vehículos se encuentran a una distancia que en situación ideal permitiría la comunicación, y el vehículo de la derecha (vehículo 1) le podría enviar al vehículo de la izquierda (vehículo 2) uno de los contenidos que tiene almacenados, pero al realizar la transmisión un edificio de la ciudad se interpone entre los mismos y el flujo de datos no puede atravesar las paredes del edificio, por lo que el vehículo 2 no puede obtener el contenido.

Por otro lado, la simulación basada en agentes no es la única vía para recrear situaciones hipotéticas en las que objetos activos se mueven a través de un medio. Según [24], las opciones más comunes son las siguientes: Evento discreto, en las cuales eventos aislados desencadenan la acción y hacen que el sistema vaya variando progresivamente, dinámicas de un sistema, en el cual a través de flujos y retroalimentación varía el estado de los objetos del sistema y sistemas dinámicos, los cuales son los antecesores de las dinámicas de un sistema y se representan a partir de diagramas de bloques en campos como la electrónica o la mecánica de fluidos.

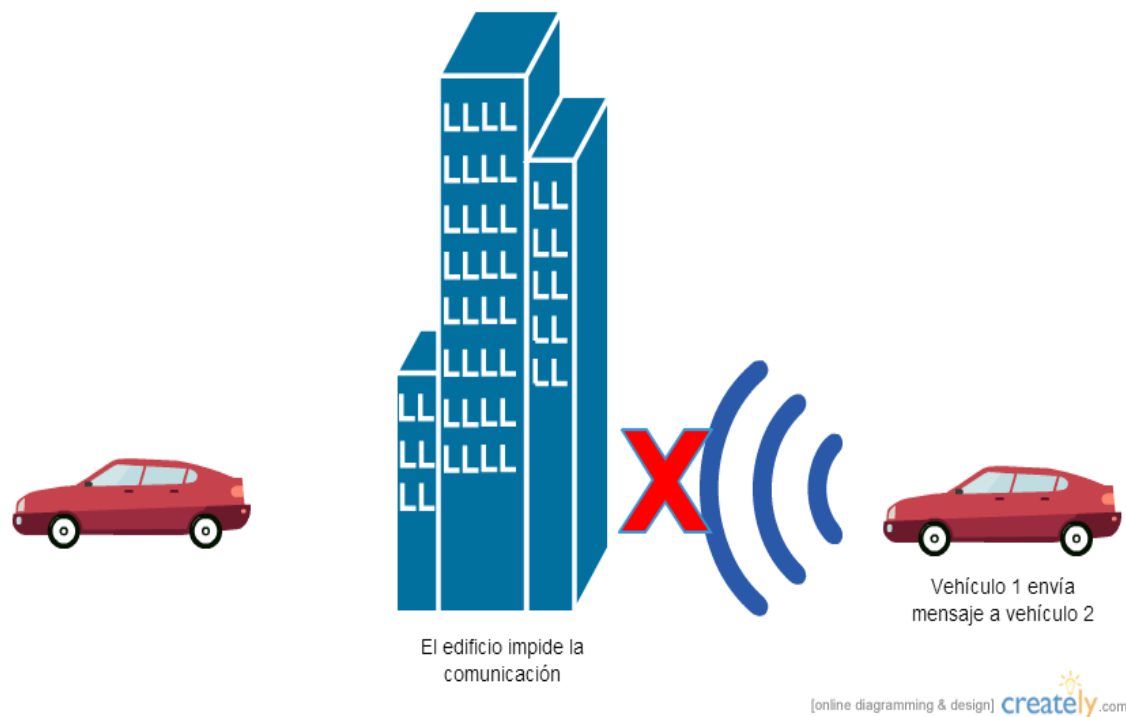


Ilustración 2-6. Interacción de los agentes con el medio en el que se encuentran

La elección de un modelo basado en agentes es su carácter general, lo que lo hace válido para la resolución de cualquier tipo de problema, además de su capacidad para tratar problemas más complejos y con más dinamismo. Este modelo también resulta muy adecuado cuando no se tiene un conocimiento sólido sobre el efecto que tiene la interacción de unos agentes con otros y con el medio, que es lo que ocurre con las redes vehiculares, pues al tratarse de un concepto tan novedoso no se tiene conocimiento a priori de cuál va a ser el resultado de utilizar un parámetro u otro.

Tras la revisión del estado del arte en las redes vehiculares se ha podido comprobar cómo la distribución de contenidos es el campo de aplicación que menos se ha estudiado, y que las propuestas que se han realizado quizá en algunos aspectos estaban incompletas, pues se habían realizado en escenarios poco realistas. Es por eso que en este proyecto se estudiarán tres protocolos exponiéndolos a situaciones en las que antes no se han probado, pudiendo extraer nuevas conclusiones. El hecho de que no se hayan implementado redes vehiculares que permitan la distribución de contenidos, y que las propuestas realizadas no aborden por completo todas las posibles situaciones refuerza la elección del modelo basado en agentes propuesto en este proyecto.

3. Distribución de contenidos: Procedimiento

La finalidad de este trabajo es la evaluación de distintos mecanismos de distribución de contenidos en redes vehiculares dentro de un entorno urbano, para así poder extraer conclusiones sobre el protocolo más efectivo. Para llevar a cabo este estudio se ha optado por simular el tráfico de una ciudad, en concreto el centro de Madrid, a través del uso de agentes autónomos. Estos agentes, además de simular el movimiento de los vehículos, también tendrán la capacidad de obtener contenidos a través de dos vías distintas. La primera, la cual será la prioritaria y es el objeto de estudio del proyecto, es obtener el contenido a través de vehículos cercanos u estaciones de información que estarán en puntos fijos de la ciudad y proveerán conexión a Internet. En caso de no obtener dicho contenido por la VANET, los vehículos descargarán el contenido a través de su conexión 3G.

Cada una de las simulaciones replicará el tráfico urbano durante un tiempo de 3600 segundos, con un tiempo medio de estancia de cada vehículo en la simulación de alrededor de 3 minutos. Se modificarán distintos parámetros como son el flujo de vehículos o la cantidad de contenidos para estudiar qué resultados tiene la aplicación de cada uno de los protocolos que se tendrán en cuenta, lo que hará que el número de simulaciones sea bastante elevado. Este hecho, junto con la intención de que este proyecto sirva como base para otros de una naturaleza más compleja, ha producido que uno de los factores que se han tenido en cuenta en el momento de la programación sea el tiempo de ejecución, intentando minimizarlo para futuros casos en los que la cantidad de operaciones que se requieran simular sea mayor.

3.1. Mapa

El mapa utilizado para realizar este tipo de simulación debe tener capacidad de abarcar un gran número de rutas distintas en las que los vehículos puedan distribuirse, pues un mapa pequeño produciría situaciones en las que los vehículos circulan por las mismas vías, haciendo la comunicación mucho más sencilla al estar todos aglomerados. Para evitar esta problemática, se ha utilizado un cuadrado del centro de Madrid de 4.5 km de lado, lo que supone un mapa lo suficientemente amplio como para que el tráfico simulado sea muy similar al real con el número de vehículos que circularán. En la Ilustración 3-1 se puede observar el escenario utilizado, obtenido de SUMO, el programa encargado de simular el movimiento de los vehículos.

El número de vehículos que se han distribuido en este mapa ha sido desde 2000 hasta 9000 para así poder generar situaciones de baja densidad de vehículos en las que la comunicación sea más difícil, y otras con una alta densidad de tráfico en las que los vehículos tengan más posibilidad de encontrar vecinos que le faciliten el contenido que está buscando.

Para el correcto funcionamiento de la VANET se han colocado en el mapa en posiciones fijas cuatro estaciones de información, las cuales se pueden ver como puntos rojos. Se

han colocado en intersecciones importantes donde circula un número mayor de vehículos, lo que favorece que el mayor número de vehículos puedan encontrar contenidos para que luego puedan diseminarlos por el resto del mapa mientras se desplazan con el fin de llegar al final de su trayecto.



Ilustración 3-1. Plano de Madrid utilizado, de 4.5 km x 4.5 km

3.2. Programas utilizados

En el caso de los agentes necesarios para realizar el estudio, necesitan moverse autónomamente por las calles de una ciudad, y la capacidad de establecer comunicación con otros vehículos para obtener el contenido que desee. Se ha optado por realizar en dos programas distintos cada una de las actividades: uno se encarga de generar los movimientos, y el otro se encarga de la comunicación. La instalación y puesta en marcha de este entorno de simulación se ha detallado en el Anexo III. Manual de instalación del entorno de simulación.

3.2.1. SUMO

El movimiento de los vehículos a través de la ciudad es un factor importante en la simulación ya que para comprobar si los protocolos de distribución son efectivos en un entorno real es necesario que estos vehículos actúen como lo harían en una situación real. Es por eso que se ha apostado por la utilización de un simulador de tráfico como SUMO (Simulator of Urban MObility) [25], el cual genera automáticamente las rutas de los vehículos y añade funcionalidades como la existencia de semáforos o cruces que producen aglomeraciones de vehículos, hecho típico en la vida real y que puede ser beneficioso para el funcionamiento de una VANET al aumentar la concentración de nodos y disminuir el dinamismo de la topología de red.

Para el correcto funcionamiento de SUMO se necesita establecer cuál va a ser la topología urbana que se va a utilizar. En este trabajo se ha utilizado un mapa del centro de Madrid, por lo que se obtuvo a través de la herramienta OpenStreetMap un archivo en formato .osm que contenía la información de las calles de la ciudad. SUMO trabaja con archivos en formato .xml, por lo que incluye una herramienta llamada NETCONVERT que convierte el archivo de formato osm a xml de manera automática⁴.

Una vez configurado el escenario en el que se van a desplazar los vehículos, el siguiente paso es crear las posibles rutas que pueden seguir los vehículos, ya que para insertar un vehículo en SUMO es necesario asignarle cuál es su punto de origen y cuál el de destino. Como la cantidad de tráfico que se desea añadir es bastante elevada, sería demasiado tedioso insertar manualmente las rutas. Para evitarlo, se ha utilizado una herramienta de SUMO llamada dualroute, que automáticamente genera las rutas siguiendo el algoritmo Dynamic User Assignment (DUA), el cual permite distribuir los vehículos entre los posibles caminos.

Se ha considerado una distancia mínima de ruta de 1km para los vehículos, ya que sería irrealista situaciones en las que los usuarios utilizaran ese medio de transporte para desplazarse distancias tan cortas, además que sería poco eficiente para la VANET al recorrer ese espacio en poco tiempo sin que le haya dado tiempo a conseguir el contenido. En cuanto a la velocidad máxima que pueden alcanzar, se ha fijado como 50 km/h, que es la permitida en ciudad según la legislación española.

3.2.2. TRACI4J

Una vez comprendido el funcionamiento de SUMO, se puede inferir el hecho de que este tipo de simulador es muy útil para la simulación del tráfico urbano, pero no ofrece la posibilidad de recrear el funcionamiento de la VANET. SUMO ofrece, por otro lado, una herramienta para la comunicación externa con el programa, para así obtener datos de la simulación. Esta herramienta es TRACI (TRAffic Control Interface), la cual establece un protocolo de comunicación con SUMO. Para la realización de este proyecto, se ha usado una aplicación a más alto nivel, TRACI4J (TRACI for Java), la cual permite, a través del lenguaje de programación Java, realizar esta comunicación con sencillas instrucciones. Aparte de TRACI4J existen otras implementaciones a alto nivel como TRACI4Matlab, o usando Python (lenguaje por defecto con TRACI). La elección de TRACI4J viene determinada por el simulador que se expondrá a continuación, el cual trabaja también con Java, y facilita la interacción entre las dos aplicaciones.

⁴ Posteriormente a la generación automática, es necesaria la edición manual del documento para subsanar errores, como por ejemplo intersecciones que no se hayan representado correctamente, o vías que no estén bien conectadas. Dicho trabajo de edición no se ha realizado en este trabajo, sino que ha sido facilitado por los autores de [9].

3.2.3. MASON

La espina dorsal de todo el sistema se encuentra en MASON (Multi-Agent Simulator Of Networks) [26], el cual será el utilizado para ensamblar todos los componentes y dotar de capacidad de comunicarse a los vehículos. MASON es una librería para Java que permite realizar simulaciones basadas en agentes autónomos dentro de cualquier ámbito, como pueden ser la simulación de la evolución de una colonia de hormigas o el movimiento del Sistema Solar, ejemplos que ofrecen sus creadores para ilustrar su funcionamiento.

Para comenzar a utilizar MASON se puede descargar desde su página web un manual en el que se explica detalladamente cómo funciona, proponiendo ejemplos paso por paso para la comprensión del código. Es necesario para su ejecución la instalación de algunas extensiones que permiten la visualización de simulaciones en 3D, generación de documentos en PDF o películas. Además, para la realización concreta de este trabajo se requiere la extensión GEOMASON, la cual permite añadir figuras geométricas a MASON, lo cual será útil para la visualización de los edificios.

Por otro lado, esta librería incluye también la opción de la visualización de la simulación, lo que la hace aún más interesante, pues se puede comprobar de forma visual si los movimientos y las interacciones están siendo los adecuados o no. En la Ilustración 3-2 se puede observar esta visualización durante las simulaciones realizadas desde un plano general, que es el que muestra inicialmente MASON cuando se comienza a ejecutar. Concretamente en la parte izquierda, la distribución de los edificios es la misma que en la imagen del plano de Madrid, aunque con menos detalle. En la parte derecha aparecen los vehículos representados como puntos de color gris.

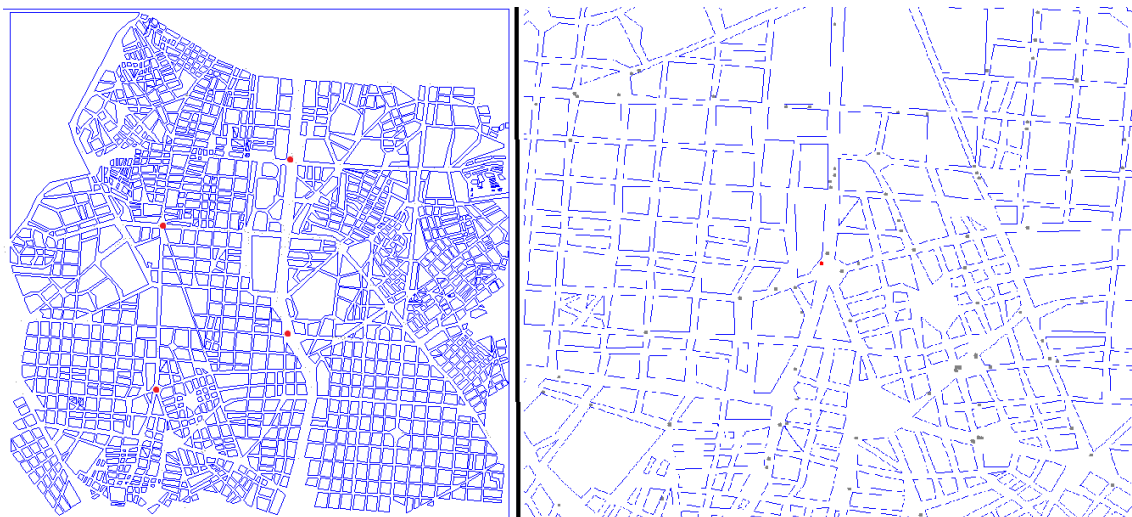


Ilustración 3-2 Plano de Madrid visto en MASON (Izquierda) y plano aumentado mostrando los agentes (derecha)

Haciendo click sobre los vehículos se puede obtener información como el identificador de vehículo o si han encontrado el contenido a través del campo contentsReq. Si esta variable tiene el valor -1 indica que todavía no han comenzado a buscar ningún contenido, valores mayores que 0 indican que el vehículo está buscando el contenido con ese indicador, y si el valor es 0, ya han obtenido ese contenido a través de la VANET, como

es el caso del vehículo que se ha seleccionado en la Ilustración 3-3, donde además se puede ver que el identificador del vehículo es el 1449, lo que permite diferenciarlo del resto. Si lo han hecho a través de la descarga por 3G, este valor será -2. Por otro lado en el campo contents se pueden observar los contenidos que tiene almacenado en su memoria caché.

Por otro lado, representadas como puntos de color rojo aparecen las estaciones de información, las cuales tienen un comportamiento totalmente distinto al de los vehículos al permanecer estáticas. Estas estaciones, al ser las que poseen conexión a Internet, poseen todos los contenidos que les solicite cualquier vehículo cercano.

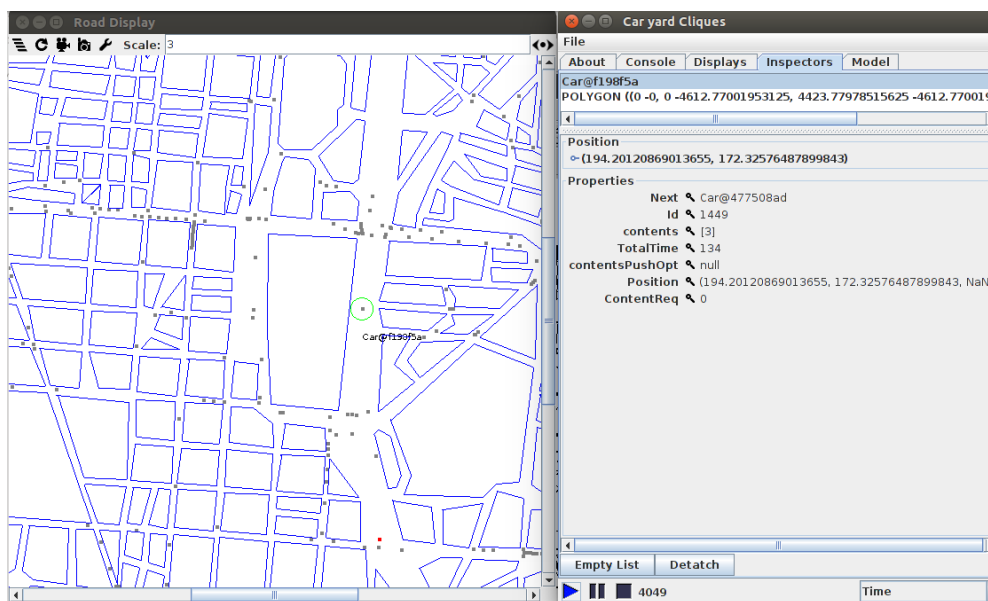


Ilustración 3-3. Selección de datos de un vehículo

3.2.4. Obstáculos

Desde el punto de vista de la simulación, los edificios del mapa serán considerados como obstáculos para la comunicación, ya que si dos vehículos no comparten visión directa porque existe un edificio que lo impide, la comunicación no será posible.

SUMO no facilita la información sobre los edificios del mapa en el que se desplazan sus vehículos, por lo que se ha tenido que crear un archivo aparte donde aparezcan las coordenadas de cada uno de los edificios para que posteriormente MASON obtenga esas coordenadas y cree los obstáculos dentro de la simulación. Además, estas coordenadas son necesarias para que el simulador pueda mostrar en el modo de visualización estos edificios para poder comprobar que los vehículos están circulando por las carreteras sin entrar dentro de los edificios, lo que implicaría un problema de escala.

La elaboración del archivo de obstáculos se ha realizado manualmente⁵ configurando polígonos cerrados con las distintas coordenadas que se fijaban. En la mayoría de los

⁵ La creación del archivo no se ha realizado en este proyecto, sino que ha sido facilitada por los autores de [9].

edificios las coordenadas no coinciden exactamente con las que se pueden extraer de OpenStreetMap para incluir así las aceras de las calles que no serían un obstáculo para la comunicación pero sí están incluidas dentro de las coordenadas de los edificios.

3.2.5. Diagrama general

Por último, se mostrará a través de un diagrama (Ilustración 3-4) cuál es el funcionamiento general de las simulaciones a través de todos los programas que se han explicado anteriormente.

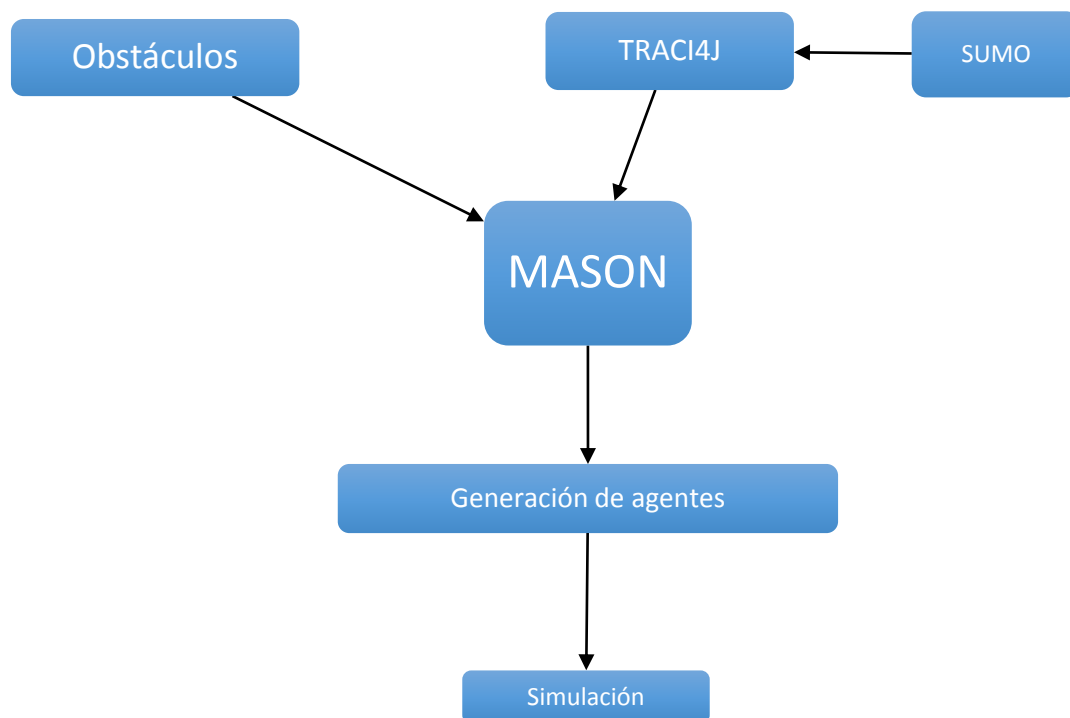


Ilustración 3-4. Componentes del entorno de simulación

Como se ha explicado anteriormente, MASON es el que ensambla todas las partes de las que se han hablado en los distintos apartados anteriores. Inicialmente, cuando se ejecuta la aplicación se eligen los parámetros de configuración de la simulación como se puede comprobar en la Ilustración 3-5, y una vez elegidos, se inicializa a través de MASON el escenario en el que se va a realizar la misma. Los principales parámetros que se configuran son el tamaño de la ventana de visualización, los colores con los que se representarán los agentes, y la inicialización de todas las variables que podrán ser utilizadas. Por otro lado, también se hace una llamada a la ejecución de SUMO utilizando TRACI4J, indicando dónde se encuentran los archivos necesarios para realizar la simulación del tráfico. En ese momento SUMO realiza la carga del mapa que se ha especificado e inicializa la simulación. También es necesario cargar el mapa de obstáculos para que esté listo una vez que comience la circulación de vehículos, por lo que se crean

los objetos que representan los obstáculos y se realiza la representación en MASON para su visualización.

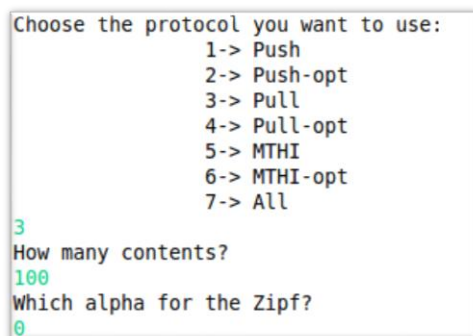


Ilustración 3-5. Menú de selección de parámetros

En el momento en que se han configurado e inicializado los dos simuladores, se procede a la ejecución de la simulación. En primer lugar se añaden a la simulación las estaciones de información que proveerán de contenidos a los vehículos cercanos, las cuales no necesitarán actualización de posición y se mantendrán fijas desde el primer momento. Posteriormente, SUMO añade periódicamente vehículos al escenario para que recorran la ruta que tienen asignada. En cada paso de la simulación, MASON realiza una petición a SUMO a través de TRACI4J para que le envíe la posición de los vehículos que hay en el escenario, y en consecuencia, MASON creará un nuevo agente si ese vehículo acaba de entrar a la simulación, o actualizará sus coordenadas si ya había aparecido en un paso de la simulación anterior. En el momento en que se crea en MASON un agente, y hasta que deja de recibir información del mismo (en el momento en el que ha alcanzado su destino y se elimina de la simulación de SUMO), el agente vehículo tiene la capacidad de pedir contenidos y servirlos al resto si los posee en función del protocolo que se esté simulando en ese momento.

Cuando se alcanza el fin de la simulación, es decir, a los 3600 segundos, MASON ordena la finalización de la conexión a SUMO utilizando TRACI4J, y realiza un volcado de los resultados en archivo de texto plano indicando cuáles han sido los parámetros de la simulación para poder diferenciarlos. Los datos están tabulados entre sí para su posterior utilización en una hoja de cálculo, lo que facilita su representación y visualización.

3.3. Estructura interna y agentes

En el apartado anterior se ha comentado el funcionamiento del simulador desde un punto de vista externo, sin entrar en detalles sobre cómo funcionan los agentes y la simulación por dentro. En este apartado se profundizará en la estructura interna de la simulación, haciendo hincapié en los aspectos más técnicos.

Para la utilización de MASON se requiere la creación de distintos bloques que se encarguen de diferentes aspectos de la simulación. Esta especialización en bloques permite la separación de tareas, y hace más sencilla la identificación de errores, y los

cambios modulares, es decir, al cambiar internamente un módulo no afecta al resto de módulos, pues pertenecen a partes distintas de la simulación.

El primer bloque es el escenario de la simulación, el cual es el lugar en el que se sitúan los vehículos, las estaciones de información, los obstáculos de la ciudad, además de toda la información relativa a la simulación, como qué protocolo se está usando, cuántos vehículos, o si existe o no popularidad. El escenario es el bloque común de toda la simulación, donde el resto de bloques pueden extraer sus parámetros necesarios para funcionar.

El segundo bloque es opcional, y es el de la visualización. Durante el grueso de las simulaciones se ha omitido la visualización de las mismas para optimizar el tiempo de ejecución, debido a que la representación de cada uno de los agentes en cada instante de tiempo consume numerosos recursos. El bloque de visualización recoge la información del primer bloque, el del escenario, y se encarga de dar color a los agentes y a mostrarlos a ellos y a los obstáculos en el mapa.

El tercer bloque está compuesto por los agentes. Se pueden distinguir asimismo dos tipos de agentes, los vehículos, y las estaciones de información. Los vehículos reciben periódicamente una actualización de su posición, y aparecen durante un tiempo limitado en la simulación, mientras que las estaciones se mantienen fijas desde el comienzo hasta el fin de la simulación. Las estaciones son las encargadas de insertar los contenidos en la red, ya que sin ellas todos los vehículos comienzan sin contenidos en su memoria caché y tendrían que descargarlos por 3G al no existir vehículos que posean contenidos. Todos los vehículos que al comenzar la simulación se encuentren a una distancia que les permita comunicarse con las estaciones podrán obtener el contenido que desean, y al moverse por la ciudad podrán diseminarlo al resto de vehículos.

El último bloque es el de los contenidos, el cual está representado en la mayoría de protocolos por estructuras de datos sencillas (números enteros), exceptuando el protocolo *Push* con optimizaciones que exige guardar una marca de tiempo con cada contenido para identificar qué contenidos no se encuentran actualmente en las proximidades del vehículo.

3.4. Consideraciones sobre los protocolos usados

Para la realización del estudio de mecanismos de distribución de contenidos, siguiendo [9], se han elegido tres protocolos representativos que serán los que se implementarán y compararán entre ellos. Estos protocolos son *Push*, *Pull* y *MHTI* (Multi-Hop To Infostation). Cada uno de ellos tendrá la opción de implementar optimizaciones dentro de su protocolo, lo que servirá para hacer una comparativa aún más afinada.

En todos los protocolos que se han implementado se ha tenido en cuenta que el vehículo está buscando un único contenido desde el momento en que entra en la simulación, y si tras 120 segundos no se ha conseguido el contenido deseado a través de la VANET, éste se descarga por 3G. Una vez obtenido el contenido por cualquiera de las dos vías, el vehículo ya no busca ningún contenido más. Por otro lado, la memoria de almacenamiento

de contenidos es infinita, por lo que pueden almacenar cualquier cantidad de contenidos. La distancia de cobertura radio que se ha considerado entre dos nodos es de 200 metros con visión directa, cobertura típica de las tecnologías 802.11 en la banda de 2,4 GHz.

En este modelo de simulación no se consideran parámetros físicos como la atenuación en distintos medios, y no se contempla un posible error de transmisión o recepción, solamente se considerará la distancia y la no existencia de un edificio en el campo directo de visión a la hora de considerar una transmisión válida. Tampoco se considera la posibilidad de colisiones si varios nodos transmiten al mismo tiempo, lo que se justifica porque todos los protocolos que vamos a estudiar generan una carga muy baja en la red.

Se distinguirán a la hora de realizar la simulación los mensajes que incluyan contenidos y los que simplemente realizan una petición o envían una coordenada. Los primeros tendrán un tamaño de 1 Kilobyte, mientras que los otros verán su tamaño decrementado a una décima parte, 100 bytes. Este dato tendrá especial utilidad para medir la sobrecarga de la red, en la que se tendrán en cuenta la relación entre los bytes de contenido deseado conseguidos a través de la VANET y el total de bytes enviados.

Para darle una mayor fiabilidad a los resultados de estas simulaciones, se ha realizado cada experimento 5 veces usando 5 rutas de tráfico distintas. Con estas 5 muestras se utilizará la media de las mismas como valor definitivo, y se mostrará el intervalo de confianza que se puede obtener a partir de una distribución t de Student de parámetro 4 (grados de libertad) y un alfa de 0.05 para generar niveles de confianza del 95%.

El protocolo *Push* (Ilustración 3-6) se caracteriza por transmitir contenidos en *broadcast* sin previa petición de los mismos. El agente (que puede ser tanto un vehículo como una estación) envía cada 10 segundos un contenido que tenga almacenado en su memoria caché, elegido de forma aleatoria. Si ese contenido es recibido por un vehículo que estaba buscando el mismo, este lo envía a la aplicación automáticamente, lo cual no implica que se deje de transmitir contenidos cada 10 segundos.

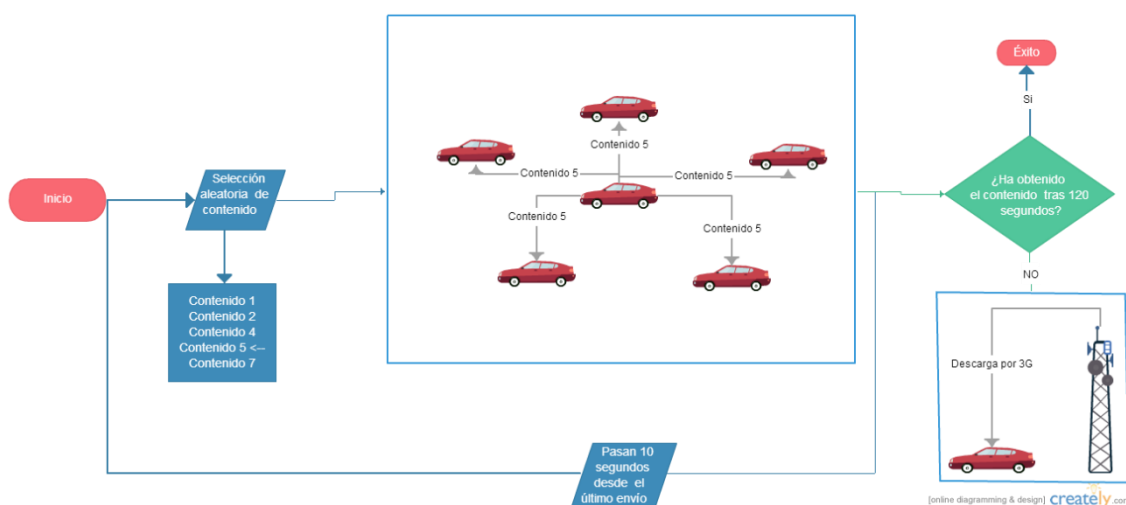


Ilustración 3-6. Funcionamiento de Push

La versión optimizada de *Push* (Ilustración 3-7) mejora la selección del siguiente contenido a transmitir. Para evitar que un mismo contenido sea transmitido varias veces en la misma zona (lo que es poco eficaz porque los vehículos ya lo tendrán almacenado), la elección del contenido no es aleatoria sino basada en una marca de tiempo que tiene cada contenido almacenado, y que va incrementándose cada segundo. El contenido elegido será aquel cuya marca de tiempo sea mayor, y si existen varios contenidos con idéntica marca de tiempo, se elegirá aleatoriamente uno de ellos. Tras la transmisión esta marca tendrá valor 0 para evitar que el contenido vuelva a ser elegido en un instante cercano. Cuando un vehículo reciba un contenido que ya tenga almacenado en su caché, le cambiará la marca de tiempo también a 0, ya que volver a emitir este contenido sería inútil pues sus vecinos también lo habrán recibido al mismo tiempo que él. Por último, si un vehículo descarga por 3G un contenido, este será el elegido para la transmisión inmediatamente posterior, ya que el hecho de que se haya descargado por 3G implica que ese contenido no estaba presente en esa zona geográfica.

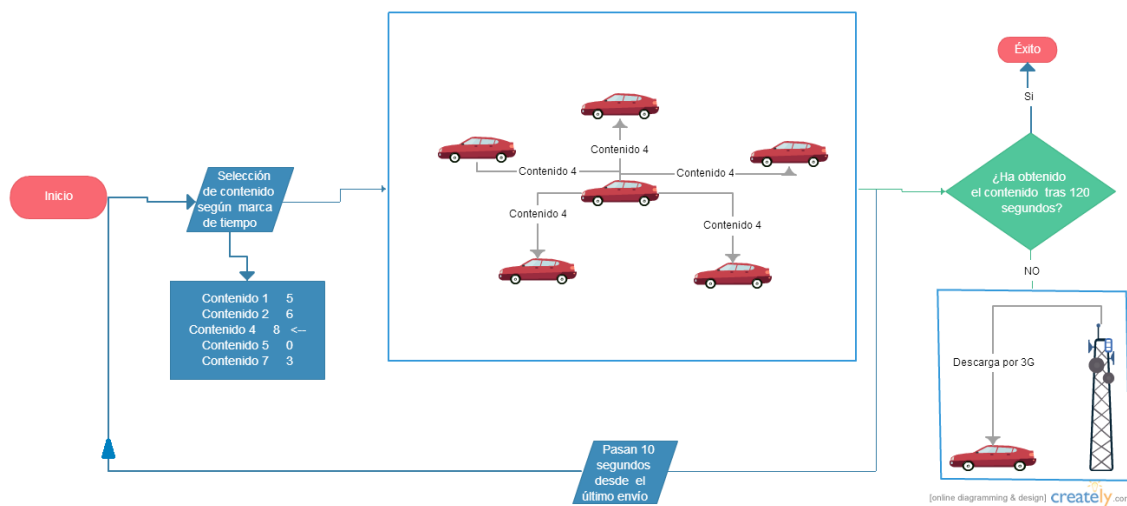


Ilustración 3-7. Funcionamiento de *Push* optimizado

En cuanto al protocolo *Pull* (Ilustración 3-8), se diferencia con el primero porque éste no transmite en *broadcast* un contenido de su caché, sino que cada 10 segundos lanza un mensaje de petición del contenido que está buscando, y si alguno de los vehículos o estaciones que han escuchado el mensaje lo tienen, se lo envían. Si a los 120 segundos de haber iniciado la búsqueda del contenido no lo ha recibido, el vehículo lo descarga a través del 3G, y ya no vuelve a realizar peticiones. A partir de ese momento, el vehículo continúa en circulación y si algún vecino le realiza la petición de un contenido, se lo envía.

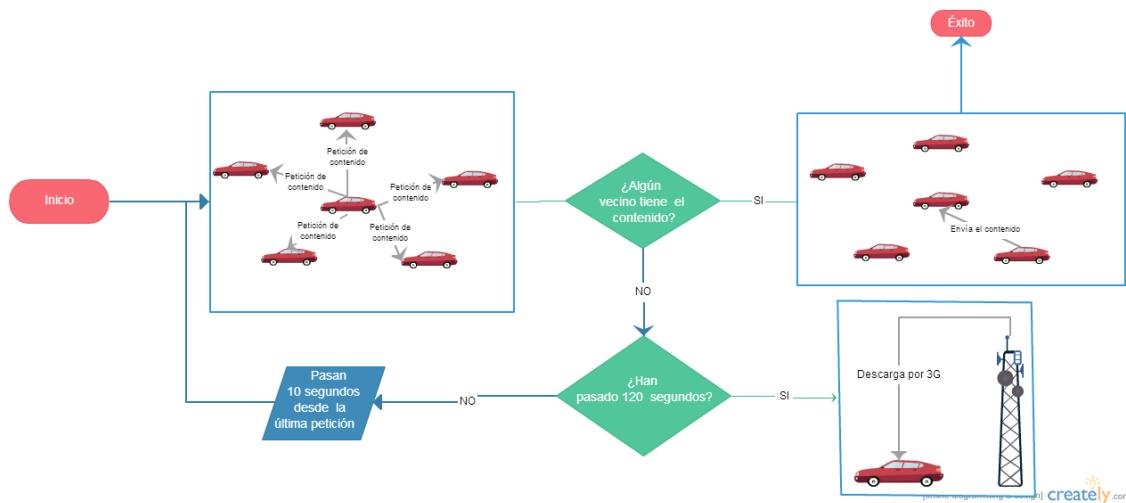


Ilustración 3-8. Funcionamiento de Pull

La versión optimizada del protocolo *Pull* (Ilustración 3-9) realiza una mejora en la sobrecarga de la red, ya que los vehículos al recibir una petición de un contenido que ellos poseen escuchan si hay un vehículo más cercano al emisor que envía el contenido, y en este caso no se realiza el envío para evitar duplicidad de envíos. Además, los contenidos se envían en *broadcast* para que vecinos cercanos que puedan necesitar el mismo contenido no necesiten pedirlo. En el caso de que un vehículo se descargue por 3G el contenido, éste también será enviado a todos sus vecinos dado que si no lo ha conseguido a través de la VANET, el resto tampoco lo tendrán almacenado y podrá serles útil.

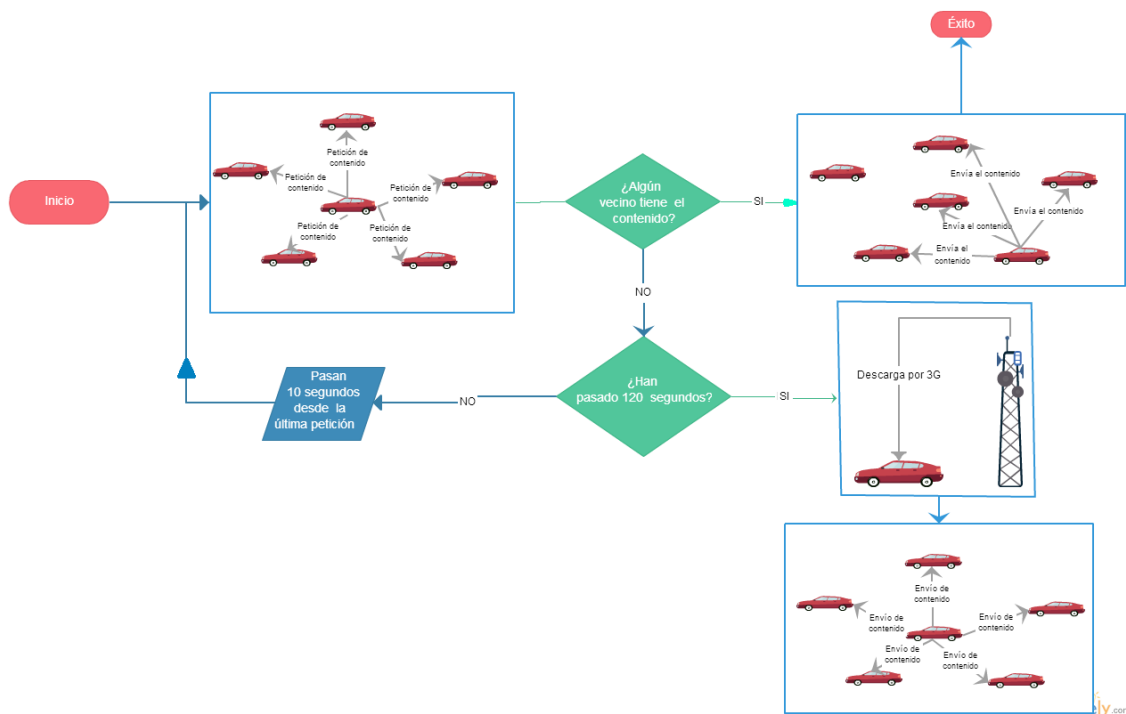


Ilustración 3-9. Funcionamiento de Pull optimizado

El protocolo *Pull* anteriormente comentado basa su funcionamiento en la búsqueda de vecinos cercanos que puedan tener el contenido que necesita, lo que resulta altamente

ineficiente si hay un gran número de contenidos, pues la probabilidad de que un nodo vecino posea el contenido es muy baja. El protocolo que se tratará a continuación, *MHTI* (Ilustración 3-10), es una evolución del protocolo *Pull*, pues realiza peticiones periódicas de contenidos cada 10 segundos, con la diferencia de que no solamente se ciñe a conseguir el contenido de sus vecinos. Cuando realiza la petición de un contenido, los nodos vecinos lo envían si lo tienen almacenado en su caché, y en caso contrario, si se encuentran más cerca que el vehículo emisor de una estación de información, envían sus coordenadas geográficas. El vehículo que en primer momento ha pedido el contenido, en caso de no recibirlo en este primer intento, selecciona entre todas las coordenadas que ha recibido la del vehículo que se encuentre a menor distancia de su estación de información (donde el contenido estará disponible) más cercana, la cual tendrá una posición conocida. Una vez seleccionado le envía un mensaje donde se le informa de que es él quien tiene que continuar la búsqueda, es este el que repite el proceso, y así de forma iterativa hasta que se encuentra una estación de información, uno de los vehículos vecinos posee dicho contenido, o no se encuentran más vecinos que puedan seguir reenviando la petición hacia la estación de información. Si finalmente se ha alcanzado el contenido, éste se va enviando siguiendo el camino inverso que se ha seguido durante la ida, y los vehículos que lo reenvían al mismo tiempo lo almacenan en su caché.

Ilustración 3-10. Funcionamiento de MHTI

Como se ha explicado en este capítulo, el entorno de simulación que se ha implementado goza de un gran nivel de especialización, separando la función de movilidad de los vehículos en un programa y la realización de las comunicaciones en otro. Esto es muy lógico, ya que como se ha presentado en el estado del arte, la simulación basada en agentes permite estudiar comportamientos autónomos de los cuales no se posee demasiada información, algo que de hecho ocurre con las comunicaciones en las VANETs. En cambio, el estudio del tráfico urbano y los patrones específicos que siguen los conductores es un tema sobre el que se ha trabajado durante décadas, por lo que el uso de un simulador con la reputación de SUMO se convierte en la mejor elección. Por otro lado, han sido elegidos un mapa y unas condiciones de simulación que puedan adaptarse lo más fielmente posible a una situación real, utilizando el centro de la ciudad de Madrid con sus edificios reales como obstáculos, y un tiempo de simulación y unas densidades de tráfico que se han estimado que pueden reflejar con exactitud una comunicación real.

Los tres protocolos que se han implementado también pretenden estar diseñados para satisfacer las necesidades de los usuarios que utilizarían la VANET para consumir contenidos, con unos tiempos máximos de obtención del contenido y un tamaño de contenido que se han estimado que serían los utilizados una vez puesta en marcha la tecnología.

4. Resultados de las simulaciones

En este apartado se mostrarán los resultados principales de las simulaciones que se han realizado, divididos en subapartados en los que se irán variando parámetros de la simulación para comprobar cómo evoluciona cada protocolo.

4.1. Densidad del tráfico

El primer parámetro que se analizará en este entorno de simulación será la influencia que tiene la densidad de tráfico en el comportamiento de la VANET. El tráfico mínimo considerado será de 2000 vehículos, que corresponde a una densidad de tráfico muy baja en el centro de una ciudad, e irá aumentado de 1000 en 1000 hasta llegar a 9000 vehículos, una densidad de tráfico alta sin saturación. Para la correcta interpretación de los datos, es necesario indicar de nuevo que los vehículos tienen 120 segundos para conseguir el contenido que desean, y de no ser así lo descargarán por 3G. En este primer experimento el número de contenidos estará reducido a 1, por lo que todos los vehículos pedirán el mismo contenido.

En la Ilustración 4-1 se muestra la relación entre la densidad de vehículos en la red y la tasa de éxito obtenida en media, que es el tanto por ciento de contenidos conseguidos a través de la VANET (el resto se habrán conseguido por 3G) o también, al buscar los vehículos un solo contenido, la proporción de los mismos que lo han obtenido usando la VANET. Se puede observar que apenas existe diferencia entre un protocolo y otro, ya que la probabilidad de que uno de los vecinos contenga el contenido deseado es muy alta al existir en la red un único contenido que puede ser solicitado.

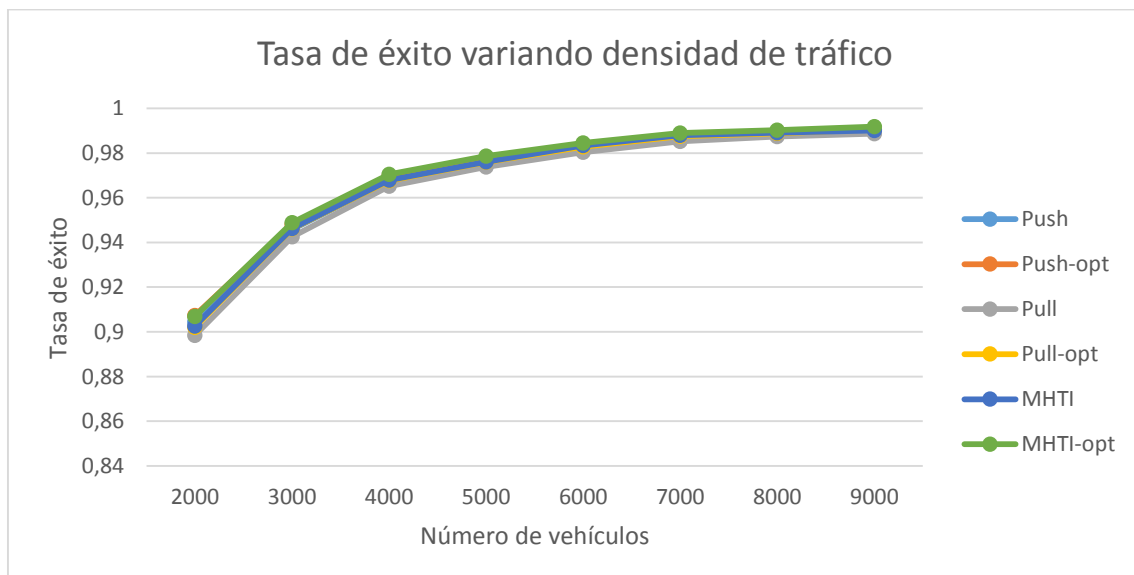


Ilustración 4-1. Tasa de éxito variando densidad de tráfico (1 contenido)

En cuanto a los valores de la tasa de éxito, se puede observar que con una densidad baja de tráfico (alrededor de 2000 vehículos), esta ronda el 90% en todos los protocolos, incrementándose conforme aumenta la densidad llegando a un punto cercano al 99% con

9000 vehículos. Realmente se puede asumir que con 9000 vehículos la tasa de éxito es del 100% debido a que ese 1% que no lo consigue es debido a que la simulación se para a los 3600 segundos y quizá algunos vehículos han entrado al escenario de simulación unos segundos antes, lo que impide que les dé tiempo a obtener el contenido en tan corto espacio de tiempo.

El intervalo de confianza de cada uno de los valores que se han mostrado en la gráfica es demasiado pequeño en todos los casos como para poder verlo en la gráfica, ya que en ningún caso supera el 1% del valor de la media estimada, es más, la media de todos los intervalos de confianza es de 0,3%.

En este análisis en el que varía el número de vehículos es necesario acudir a la Tabla 4-1 para observar diferencias significativas entre algunos protocolos. La tabla muestra la sobrecarga de la red, entendiendo ésta como la relación del número de bytes de contenido obtenidos exitosamente en la VANET y el número total de bytes enviados, por lo que una sobrecarga de 1 indicaría que para obtener un contenido solo se ha realizado un envío sin una solicitud previa del mismo. En la tabla también se muestran los valores del intervalo de confianza generado para cada una de las simulaciones, que como se podrá apreciar ninguno es un valor elevado que pueda llamar la atención

Coches	Push	Push-opt	Pull	Pull-opt	MHTI	MHTI-opt
2000	26,26 ± 0.15	26,30 ± 0.15	1,92 ± 0.02	1,52 ± 0.01	2,47 ± 0.01	2,93 ± 0.02
3000	25,70 ± 0.24	25,71 ± 0.25	2,04 ± 0.03	1,38 ± 0.01	2,46 ± 0.04	2,68 ± 0.03
4000	25,72 ± 0.11	25,72 ± 0.13	2,20 ± 0.02	1,30 ± 0.01	2,53 ± 0.02	2,53 ± 0.01
5000	26,04 ± 0.18	26,03 ± 0.19	2,36 ± 0.02	1,25 ± 0.01	2,65 ± 0.03	2,44 ± 0.03
6000	26,08 ± 0.23	26,08 ± 0.25	2,57 ± 0.04	1,22 ± 0.01	2,82 ± 0.05	2,38 ± 0.01
7000	26,33 ± 0.17	26,32 ± 0.17	2,81 ± 0.02	1,19 ± 0.01	3,04 ± 0.02	2,33 ± 0.02
8000	26,69 ± 0.24	26,68 ± 0.24	3,01 ± 0.06	1,18 ± 0.01	3,22 ± 0.05	2,31 ± 0.01
9000	26,91 ± 0.18	26,90 ± 0.18	3,29 ± 0.05	1,16 ± 0.01	3,48 ± 0.05	2,28 ± 0.02

Tabla 4-1 Sobrecarga de la red variando la densidad de tráfico

Como es lógico suponer, los protocolos del tipo *Push* tendrán una sobrecarga mucho más elevada debido a que transmiten un contenido constantemente cada 10 segundos, independientemente de si han obtenido su contenido o no, lo que produce que en el cómputo global se hayan transmitido a la red varios contenidos mientras que solo se ha obtenido uno. En el caso de la optimización de Push la sobrecarga es prácticamente idéntica a la versión normal debido a que al existir un único contenido en la red, a pesar de que el algoritmo de elección de contenido es más eficiente en la versión optimizada,

esto no afectará al comportamiento del protocolo porque solo hay un contenido que se pueda elegir.

En cuanto a los protocolos *Pull* y MHTI, en este caso no se encuentran diferencias significativas debido a que como la mayoría de vecinos tiene el contenido deseado, no es necesario dar varios saltos para conseguir ese contenido, por lo que MHTI actuaría de manera muy similar a *Pull*, aunque sí se puede distinguir una pequeña diferencia entre su sobrecarga por los casos aislados en los que se hayan realizado varios saltos. En el caso de las optimizaciones, la sobrecarga es aún más baja porque ahorran la transmisión de contenidos al enviar en broadcast las respuestas a todos sus vecinos, excepto en el caso de MHTI con una densidad baja, en el cual intenta realizar más saltos y no alcanza la RSU, por lo que se descarga el contenido y lo envía a sus vecinos, produciendo un ligero aumento de la sobrecarga.

4.2. Número de contenidos

El segundo parámetro que se variará, teniendo en cuenta una densidad de tráfico media de 6000 vehículos (que es la densidad que resulta del punto en el que se alcanza una conectividad que permite obtener buenos resultados, como se puede ver en la Ilustración 4-1), será el número de contenidos que circularán por la red. En el primer análisis se tomaba un único contenido, lo que es poco realista en un entorno real, ya que cada usuario tendrá unas necesidades distintas y pedirá contenidos de diferente índole. Es por eso que se ha ido aumentando el número de contenidos en múltiplos de 10, barriendo desde 1 contenido (que está incluido en el apartado anterior) hasta los 100.000 contenidos, escenario en el cual será difícil para un vehículo obtener el contenido, pues su vecino no tendrá por qué tener ese mismo.

En la Ilustración 4-2 aparece representado el número de contenidos en el eje de abscisas y la tasa de éxito (tanto por ciento de contenidos que se consiguen por la VANET) en el eje de ordenadas. Como se puede observar, al contrario que en el caso anterior, la tasa de éxito de los protocolos cae abruptamente al aumentar el número de contenidos. En el caso de *Push*, con 100 contenidos la tasa de éxito cae al 20%, y con más contenidos es prácticamente 0 debido a la escasa probabilidad de que un vecino cercano le envíe el contenido que desea. En su optimización, las prestaciones mejoran ligeramente, pero igualmente a partir de 1000 contenidos su tasa de éxito es prácticamente nula.

En el caso de *Pull*, aunque inicialmente cae con mayor velocidad, con 100 contenidos comienza a estabilizarse en valores próximos al 22%, debido sobre todo a los vehículos que pasan cerca de las estaciones de información donde siempre podrán obtener su contenido. Utilizando la versión con optimizaciones la caída es notablemente más suave, pero a partir de los 1000 contenidos su comportamiento es muy similar.

El MHTI es de los 3 protocolos el que tiene unas mejores prestaciones al tener una caída muy suave que se estabiliza en el 55% a partir de 1000 contenidos. Esto se explica porque con la densidad de tráfico fijada, es mucho más probable que a través de varios saltos alcance la estación de información o algún vehículo vecino de uno de los nodos lo tenga

almacenado. Utilizando las optimizaciones se puede observar una ligera mejoría, mas al igual que en los anteriores protocolos, a partir de 1000 contenidos se vuelve indistinguible.

Los intervalos de confianza de todas las medias estimadas rondan el 1% o son incluso menores.

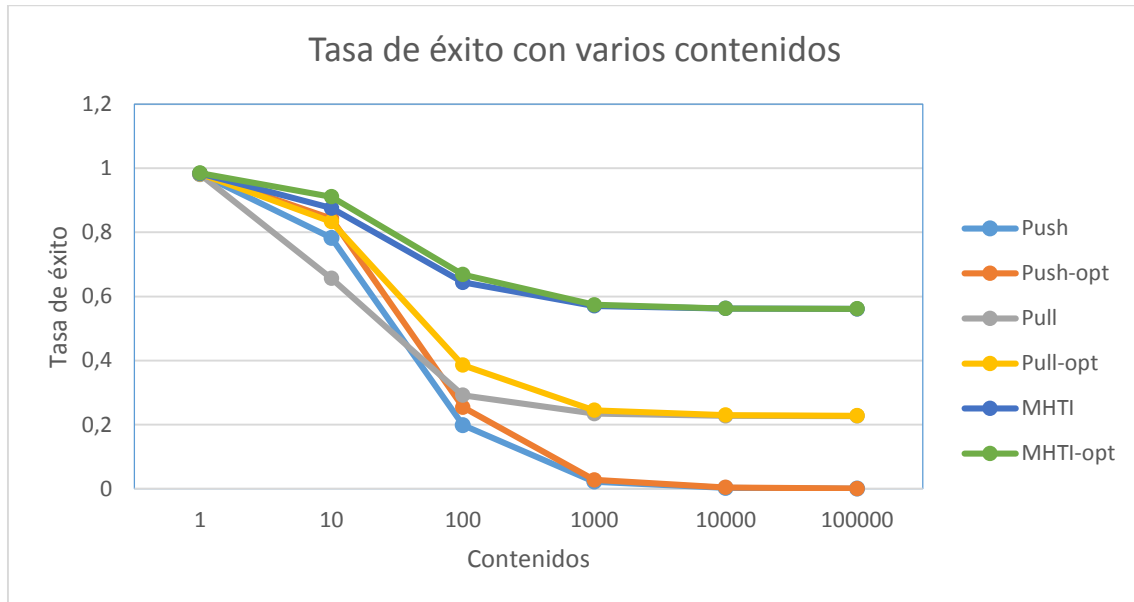


Ilustración 4-2. Tasa de éxito con múltiples contenidos (6000 vehículos)

La sobrecarga de la red se puede observar en la Tabla 4-2. En el caso de los protocolos de tipo *Push*, conforme el número de contenidos aumenta la sobrecarga tiende hacia infinito debido a que para su cálculo se divide el número de bytes enviados entre los que han sido exitosos. Se ha observado en la Ilustración 4-2 cómo los contenidos que se obtenían por la VANET tendían a 0, lo que explica esta enorme subida de la sobrecarga en la red. Los intervalos de confianza también aumentan notablemente conforme aumenta la sobrecarga, haciéndose al final realmente significativos. Esto también es debido a la forma de calcular la sobrecarga, pues al estar el número de contenidos obtenidos por la VANET en el denominador y con múltiples contenidos en la red este número es muy bajo, cualquier cambio en este valor afectará enormemente al valor de la sobrecarga.

En los protocolos de tipo *Pull* la sobrecarga se mantiene en niveles menos elevados debido a que la tasa de éxito no llega a ser tan baja. Se puede apreciar que con 1 contenido la sobrecarga en *Pull* es tres veces mayor que en su versión optimizada, mientras que con 10 contenidos no es una diferencia tan abrupta. Esto se debe a que en la optimización se escucha si algún vecino ya ha transmitido el contenido para evitar duplicidades, y con 1 contenido muchos de los vecinos tienen el contenido solicitado, lo que produce esa diferencia. Cuando el número de contenidos aumenta a partir de 1000, en la versión optimizada la sobrecarga es mayor que en la normal ya que tras la descarga de contenidos por 3G (que realiza el 80% de los vehículos) se envía en broadcast el contenido (y esto no está contribuyendo a que se consigan más contenidos por la VANET tal como se ve en la Ilustración 4-2).

El caso de MHTI es muy similar al de *Pull* en los casos particulares que se han comentado, con la diferencia de que la sobrecarga es notablemente mayor debido a que se intenta obtener el contenido a través de saltos que consumen más recursos de la red, tanto si se obtiene el contenido como si no. En todos estos casos el intervalo de confianza es muy bajo debido a que este tipo de protocolos son más estables al tener una sobrecarga mucho más pequeña.

Contenidos	Push	Push-opt	Pull	Pull-opt	MHTI	MHTI-opt
1	26,08 ± 0,24	26,08 ± 0,24	2,56 ± 0,04	1,22 ± 0,01	2,81 ± 0,05	2,38 ± 0,01
10	32,73 ± 0,18	30,41 ± 0,15	2,21 ± 0,02	1,78 ± 0,03	4,62 ± 0,06	4,31 ± 0,09
100	129,28 ± 3,83	100,72 ± 1,25	4,24 ± 0,06	4,79 ± 0,06	12,18 ± 0,12	11,81 ± 0,08
1000	1153,18 ± 91,93	934,20 ± 97,76	5,17 ± 0,09	7,89 ± 0,17	15,71 ± 0,25	16,26 ± 0,30
10000	9274,23 ± 2968,84	5859,12 ± 904,05	5,30 ± 0,09	8,44 ± 0,15	16,18 ± 0,30	16,88 ± 0,29
100000	98734,20 ± 57089,4	45842,85 ± 32651,1	5,32 ± 0,10	8,53 ± 0,16	16,25 ± 0,29	16,96 ± 0,32

Tabla 4-2. Sobrecarga de la red variando el número de contenidos

Otra información interesante que se puede extraer al añadir varios contenidos a la simulación es el número de contenidos que almacena cada vehículo en media al utilizar cada uno de los protocolos, para así poder diseñar el espacio de memoria necesario para cada vehículo. Esta información se puede observar en la Tabla 4-3.

Contenidos	Push	Push-opt	Pull	Pull-opt	MHTI	MHTI-opt
1	1	1	1	1	1	1
10	9,42	9,61	1	3,14	1,75	3,27
100	44,04	53,58	1	3,64	2,42	4,01
1000	59,00	78,62	1	3,54	2,60	3,76
10000	60,95	81,61	1	3,52	2,62	3,71
100000	61,14	81,93	1	3,52	2,62	3,71

Tabla 4-3. Media de contenidos almacenados

Como es lógico, la primera fila de la tabla estará formada solamente por unos, ya que al existir un único contenido, sólo se guardará un contenido en la memoria. A partir de los 10 contenidos se puede comprobar que los protocolos que guardan más contenidos en su memoria caché son los de tipo *Push*, en los cuales se envían contenidos constantemente a la red sin petición previa, lo que produce que numerosos vehículos guarden contenidos que no necesitan. Con 100.000 contenidos, el número medio de contenidos almacenados en el protocolo *Push* optimizado es sustancialmente mayor que en el protocolo normal, debido al hecho de que no se repiten contenidos en una misma zona durante un tiempo, lo que obliga a enviar contenidos nuevos.

En el caso de *Pull*, independientemente del número de contenidos que circulen por la red los vehículos solo almacenarán el que ellos han solicitado, ya que nadie hace *broadcast* de contenidos y todo funciona bajo demanda. Al aplicar las optimizaciones del protocolo se comprueba cómo la media de contenidos no es 1, sino que aumenta quedándose estancado alrededor de los 3,5 contenidos de media debido a que los vecinos sólo reciben contenidos en broadcast cuando alguien descarga por 3G o proporciona un contenido que alguien ha pedido.

Por último, en los protocolos MHTI se observa cómo la media de contenidos es mayor que en los *Pull* porque al obtener contenidos a través de varios saltos, los vehículos que han participado en los mismos almacenan dicho contenido, lo que hace lógico que la media sea mayor.

4.3. Popularidad

En el escenario planteado anteriormente se proponía la variación del número de contenidos que podían aparecer en la red, ya que en un entorno real el número de contenidos que podrán solicitar los vehículos será variable en función de la zona geográfica y del momento concreto en el que se desarrolle la comunicación, pero siempre asumiendo que la probabilidad de elegir un contenido venía determinada por una distribución uniforme de probabilidad. En el escenario que se describe en este apartado se propone variar un parámetro que en entornos reales aparece y que no se ha tenido en cuenta hasta ahora: la popularidad de contenidos.

En la red creada para este escenario, el número de vehículos que circula por la misma será de 6000, y el número de contenidos 1000, por lo que asumir que todos contenidos tendrán la misma probabilidad de ser solicitados es poco realista, ya que lo más común es que unos contenidos tengan más popularidad que otros y por lo tanto sea más probable que un vehículo los solicite. Es por eso que para la elección de contenidos se ha optado por una distribución Zipf, caracterizada por un parámetro alfa que en caso de ser 0 se comporta como una distribución uniforme en la que todos los contenidos tienen la misma probabilidad de ser buscados por un vehículo, y al ir aumentando alfa la probabilidad de que unos contenidos sean más probables que otros aumenta. En la simulación, se ha optado por comenzar con un alfa igual a 0 (lo que supone valores ya conocidos en el apartado anterior), e ir aumentando este parámetro en pasos de 0,25 para analizar cómo influye la existencia de popularidad en la red.

En la Ilustración 4-3 se puede analizar visualmente la influencia de la popularidad en la tasa de éxito de los vehículos en la red, la cual conceptualmente debería compensar el hecho de que haya una gran cantidad de contenidos, pues los contenidos más populares actúan como los más importantes a la hora de las prestaciones. En el caso del protocolo *Push*, a partir de alfa igual a 0,5 su tasa de éxito comienza a subir ligeramente, siendo esta subida aún mayor a partir de 1,5, que implica una popularidad alta. Con el valor máximo de alfa, consigue situarse cerca del 40%. En el caso de aplicar las optimizaciones, a partir

de alfa 0,75 la tasa de éxito aumenta con respecto al protocolo simple, llegando a situarse al final en valores cercanos al 50%.

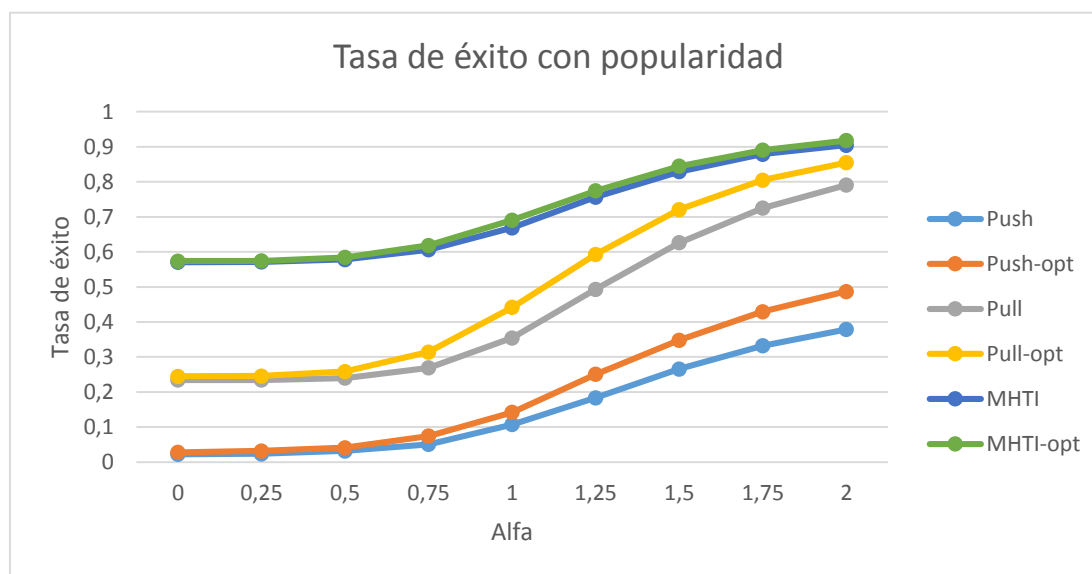


Ilustración 4-3. Tasa de éxito con contenidos con popularidad (6000 vehículos y 1000 contenidos)

En el caso de *Pull*, la mejora de prestaciones es aún más pronunciada, comenzando a subir notablemente en 0,75, aunque la subida con una pendiente más elevada sea entre 1,25 y 1,5. Finalmente se sitúa en el 80% de tasa de éxito, un valor que no conseguía alcanzar en el caso de no existir popularidad con 100 contenidos. La optimización presenta aún mejores resultados, desmarcándose prácticamente desde el inicio de su versión sencilla y situándose al final con un 85% de tasa de éxito.

Con MHTI la mejora de prestaciones no es tan acentuada debido a que las prestaciones con un alfa igual a 0 eran muy superiores al resto de protocolos. Se observa cómo la pendiente de la recta es siempre creciente, la cual aumenta notablemente a partir de 0,75, alcanzando su máximo con alfa igual a 2 por encima del 90%. En el caso de la utilización de las optimizaciones las prestaciones no se ven muy afectadas, aunque están siempre por encima de la versión normal.

El intervalo de confianza de cada uno de los valores calculados sigue siendo muy pequeño en comparación con los valores obtenidos en media, rondando en todos los casos el 0,5%.

En la Tabla 4-4 se puede observar cómo en todos los protocolos la sobrecarga de la red va disminuyendo conforme aumenta la popularidad de contenidos en la red, debido a que es más probable encontrar un contenido a través de la VANET. En el caso de los protocolos de tipo *Push*, que sin popularidad producen una sobrecarga muy alta, al final disminuyen notablemente la misma, debido a la mejora de prestaciones que se aprecia en la Ilustración 4-3. La versión optimizada está claramente diferenciada de la normal, produciendo unos valores de sobrecarga bastante menores en todas las filas de la tabla.

El protocolo *Pull* no ve variada tan drásticamente la sobrecarga debido a que este tipo de protocolo es el que genera menos de los 3 que se presentan. Sus valores van siempre

decrementándose conforme aumenta la popularidad frenando cuanto más cerca esté de alfa 2, ya que en el protocolo sin optimizaciones los vehículos no escuchan si los vecinos están transmitiendo el contenido y se dan duplicidades de envíos. En el caso de la versión optimizada se puede observar cómo esto no ocurre.

Por último, en el caso del MHTI la bajada de la sobrecarga es similar a la de los protocolos del tipo *Pull*, partiendo de la base de que el primero genera más sobrecarga debido a que realiza varios saltos.

Alfa	Push	Push-opt	Pull	Pull-opt	MHTI	MHTI-opt
0	1153,18 ± 91,93	934,20 ± 97,76	5,17 ± 0,09	7,89 ± 0,17	15,71 ± 0,24	16,26 ± 0,29
0,25	1076,09 ± 64,14	802,80 ± 28,54	5,16 ± 0,09	7,87 ± 0,17	15,68 ± 0,2	16,24 ± 0,24
0,5	808,40 ± 99,49	627,20 ± 40,97	5,06 ± 0,10	7,45 ± 0,19	15,28 ± 0,23	15,68 ± 0,32
0,75	508,56 ± 31,98	345,04 ± 24,13	4,56 ± 0,10	6,01 ± 0,17	13,89 ± 0,29	14,04 ± 0,21
1	239,55 ± 15,65	180,58 ± 2,23	3,60 ± 0,04	4,05 ± 0,03	11,03 ± 0,14	10,88 ± 0,11
1,25	139,47 ± 4,38	102,22 ± 2,94	2,78 ± 0,02	2,80 ± 0,04	7,92 ± 0,07	7,88 ± 0,19
1,5	96,53 ± 4,59	73,64 ± 1,41	2,40 ± 0,03	2,13 ± 0,03	5,84 ± 0,14	5,82 ± 0,04
1,75	77,19 ± 2,58	59,69 ± 1,2	2,28 ± 0,02	1,79 ± 0,01	4,62 ± 0,06	4,63 ± 0,06
2	67,69 ± 2,46	52,63 ± 0,85	2,26 ± 0,04	1,62 ± 0,02	4,06 ± 0,05	3,95 ± 0,08

Tabla 4-4. Sobrecarga de la red con contenidos con popularidad

Analizando el número de contenidos que almacenan los vehículos si hay popularidad (Tabla 4-5) en la red se puede deducir que un alfa mayor, es decir, con contenidos con mucha popularidad, el número de contenidos que se almacenan se reduce en buena medida, como es el caso de los protocolos *Push*, que son los que más contenidos almacenan en la red, o las optimizaciones de *Pull* y MHTI, que al enviar contenidos más repetidos en broadcast, los vehículos que los reciben ya los tienen en su memoria y no necesitan guardarlos.

Alfa	Push	Push-opt	Pull	Pull-opt	MHTI	MHTI-opt
0	59,00	78,62	1	3,54	2,60	3,76
0,25	59,09	78,47	1	3,54	2,59	3,74
0,5	58,75	77,94	1	3,54	2,58	3,78
0,75	58,30	76,68	1	3,55	2,51	3,82
1	57,32	74,69	1	3,45	2,32	3,73
1,25	56,18	72,41	1	3,19	2,05	3,38
1,5	55,12	70,65	1	2,81	1,80	2,95
1,75	54,73	69,25	1	2,45	1,61	2,55
2	54,17	68,67	1	2,16	1,46	2,22

Tabla 4-5. Media de contenidos con popularidad

4.4. Distribución de contenidos tipo Internet

En el desarrollo de los análisis anteriores se han variado parámetros que se han considerado representativos para estudiar el comportamiento de cada uno de los protocolos en situaciones que pueden ser similares a las que se dan en la realidad. En esta última sección se realizará de nuevo un estudio en función de la densidad de tráfico, pero ahora dentro de un entorno donde la distribución de contenidos sigue un modelo similar al de Internet [27].

Para seguir esta línea, se han realizado de nuevo unas simulaciones fijando el número de contenidos al máximo de todos los que se habían tenido en cuenta (100.000 contenidos) y con un nivel de popularidad similar al de Internet (alfa 0,8). Se variará el número de vehículos que aparecen en la simulación ya que es algo que realmente varía con frecuencia en una situación real, debido a que en un mismo día hay horas punta donde la densidad de tráfico es muy alta, y otras como las nocturnas en las que el flujo de vehículos es muy escaso.

En la Ilustración 4-4 se muestra la tasa de éxito en función de la densidad de tráfico. En esta situación, las prestaciones de los protocolos *Push* son prácticamente nulas, por lo que los vehículos se ven forzados a descargar su contenido utilizando la red 3G. En el caso de los protocolos *Pull*, la tasa de éxito se mantiene constante alrededor del 21% sin dejarse influir por la densidad de tráfico. Los protocolos MHTI son los que presentan unos mejores resultados, sobre todo con el incremento de la densidad de tráfico, siendo su tasa de éxito con 9000 vehículos de un 70%, una cifra significativamente más alta que en el resto de protocolos que se evalúan.

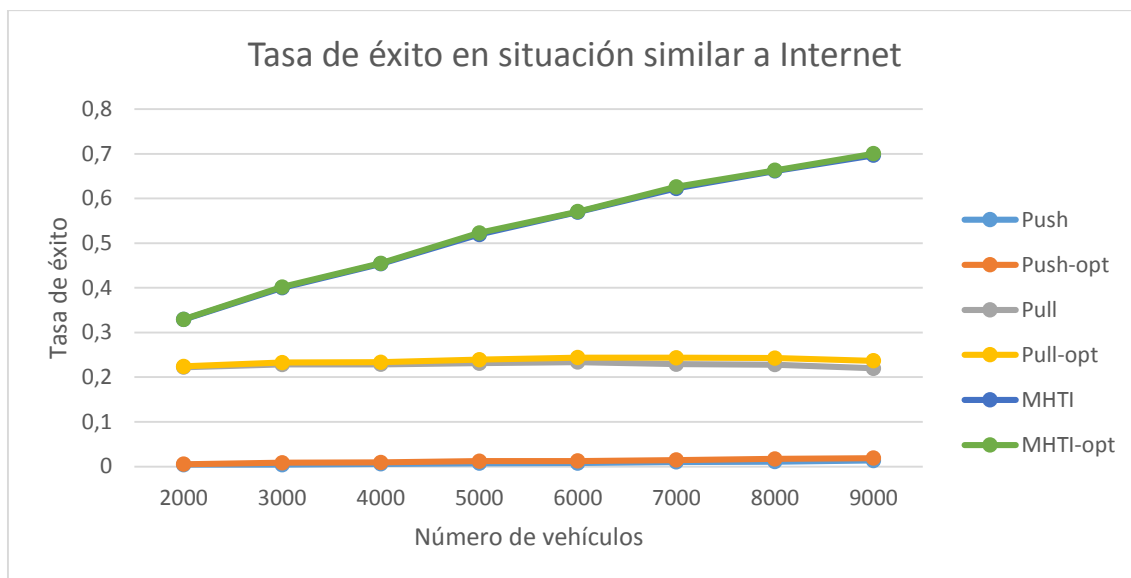


Ilustración 4-4. Tasa de éxito con $\alpha = 0.8$ y 1000 contenidos

Los valores de los intervalos de confianza siguen la tendencia de los apartados anteriores y se mantienen inferiores al 1%, sin ningún caso que presente un resultado inesperado.

Con respecto a la sobrecarga que se le inyecta a la red (Tabla 4-6), en los protocolos de tipo *Push* decremanta el valor de la misma conforme aumenta la densidad de tráfico, aunque estos valores siguen siendo muy elevados (están determinados por la tasa de éxito, que se ha visto que es muy próxima a 0). Los protocolos de tipo *Pull* tampoco se ven influidas en este caso por la densidad de tráfico, al contrario de lo que ocurre con los protocolos MHTI, los cuales van aumentando progresivamente la sobrecarga por el hecho de que tienen capacidad de realizar más saltos al existir un mayor número de vehículos a su alrededor.

Coches	Push	Push-opt	Pull	Pull-opt	MHTI	MHTI-opt
2000	5815,46 ± 1953,77	4688,44 ± 1284,01	5,42 ± 0,35	8,69 ± 0,69	14,53 ± 0,84	16,47 ± 0,76
3000	6034,34 ± 1774,68	2949,63 ± 587,41	5,31 ± 0,39	8,36 ± 0,70	14,76 ± 0,62	16,10 ± 0,69
4000	4362,86 ± 961,21	2742,14 ± 453,14	5,29 ± 0,20	8,32 ± 0,29	15,51 ± 0,44	16,58 ± 0,48
5000	3679,35 ± 1177,51	2136,83 ± 252,69	5,23 ± 0,10	8,11 ± 0,20	15,41 ± 0,35	16,14 ± 0,35
6000	3431,96 ± 1211,31	2088,88 ± 284,77	5,18 ± 0,07	7,93 ± 0,17	15,75 ± 0,28	16,40 ± 0,28
7000	2600,01 ± 524,86	1777,37 ± 138,44	5,26 ± 0,06	7,93 ± 0,17	16,15 ± 0,32	16,54 ± 0,49
8000	2481,81 ± 517,92	1566,59 ± 222,65	5,30 ± 0,15	7,97 ± 0,34	16,48 ± 0,44	16,87 ± 0,43
9000	2001,19 ± 373,82	1436,62 ± 216,62	5,47 ± 0,14	8,17 ± 0,17	17,21 ± 0,26	17,39 ± 0,18

Tabla 4-6. Sobrecarga de la red con una distribución de contenidos similar a la de Internet

Por último, en cuanto al número de contenidos almacenados en la caché (Tabla 4-7) en todos los protocolos se experimenta un aumento del número que se ha guardado, exceptuando el protocolo *Pull*, el cual como solo almacena el contenido que le interesa se mantiene estático.

Coches	Push	Push-opt	Pull	Pull-opt	MHTI	MHTI-opt
2000	22,67	28,17	1	1,82	1,33	2,01
3000	32,54	41,48	1	2,23	1,64	2,48
4000	41,31	53,87	1	2,66	1,93	2,93
5000	51,14	67,74	1	3,10	2,27	3,35
6000	60,93	81,38	1	3,53	2,59	3,73
7000	70,53	95,38	1	4,05	2,93	4,10
8000	79,63	108,37	1	4,49	3,25	4,48
9000	90,18	124,25	1	5,10	3,53	4,82

Tabla 4-7. Contenidos almacenados en un modelo similar al de Internet

5. Alternativas de diseño

Como se ha comentado en varias ocasiones, el objetivo de este trabajo es analizar mecanismos de distribución de contenidos en redes vehiculares urbanas. Este estudio se podría haber enfocado de varias formas, utilizando otras vías de simulación, o teniendo en cuenta otros factores que no se han tenido en cuenta. A continuación, se expondrán estas alternativas y se hará una breve discusión sobre por qué no se han tenido en cuenta en este trabajo.

5.1. Simulación en otras plataformas

Para simular el movimiento de los vehículos en un entorno urbano como es el centro de Madrid se optó por la utilización de SUMO debido a su carácter *open source*, lo que permite su libre utilización sin necesidad del pago previo de licencias, y además porque es uno de los simuladores de tráfico más usados en todo el mundo, utilizado en numerosas investigaciones, y con algoritmos de gestión de tráfico muy analizados satisfactoriamente. Simuladores como Quadstone Paramics o Aimsum [28] ofrecen algunas funcionalidades como visualización en 3D o manipulación del flujo de tráfico a través de herramientas gráficas que podrían resultar interesantes para un estudio más centrado en el tráfico en sí, pero para un estudio que se centra en la comunicación entre vehículos supondría una ralentización del tiempo de simulación.

A pesar de que SUMO u otros simuladores de tráfico ofrecen la opción de añadir varios tipos de vehículos como motocicletas o autobuses (en realidad SUMO no implementaría esta funcionalidad, sino una versión modificada llamada SiMTraM (Simulation of Mixed Traffic Mobility)), la inclusión de este tipo de vehículos a una simulación implicaría una complejidad de programación mucho más elevada, y llevaría mucho más tiempo su implementación, teniendo en cuenta que este estudio tiene un carácter más general en cuanto a modelos de tráfico urbano.

Para la extracción de la información de tráfico de SUMO se optó por el uso de TRACI4J, una herramienta de alto nivel programada en lenguaje Java. El uso de TRACI4J, y no otras versiones en Python o Matlab se hizo lógicamente para estar en consonancia con el simulador que manejaría los datos, MASON, el cual también utiliza este lenguaje. SUMO también facilita otra herramienta llamada TraaS (TraCI as a Service), la cual consiste en un servidor web que facilita la comunicación con Sumo utilizando TraCI, lo cual no resulta interesante al depender de conexión a internet para realizar las mismas funciones que TRACI4J. Es necesario resaltar que TraaS se probó como alternativa a TRACI4J, pero se llegó a la conclusión de que no optimizaba el tiempo de ejecución ni su implementación era más sencilla.

Otra opción a considerar a la hora de generar el tráfico podría ser aplicar en MASON algoritmos de rutas y modelos de tráfico para unificar en una única plataforma todo el proceso de simulación. Ésta alternativa presenta varios inconvenientes. El primero es que el tiempo de desarrollo del entorno de simulación se incrementa enormemente debido a

la preparación previa para la lectura y búsqueda de información de la algoritmia necesaria, y de un mayor volumen de código a programar. Por otro lado, la fiabilidad de los movimientos de los vehículos sería mucho menor, pues sería difícil llegar a la precisión de un simulador con el tiempo de desarrollo que tiene SUMO. Por último, el tiempo de ejecución podría ser mayor si no se logra una optimización del código suficiente.

En el caso de MASON, se optó por la utilización del mismo por motivos muy similares a la elección de SUMO. Es un simulador *open source*, además de ser utilizado en numerosos trabajos debido a su versatilidad y carácter general [23], algo que resultaba muy interesante debido a que la intencionalidad del trabajo es mezclar tráfico urbano y comunicación en redes vehiculares. Una alternativa a MASON que hubiera dotado al trabajo de un mayor realismo a nivel de red sería el uso de un simulador de redes como OMNET++, que ya se ha usado anteriormente en [9]. El uso de este simulador permitiría añadir de una forma sencilla parámetros como atenuación de una señal en el medio, tasas de error en la transmisión de la señal, manipulación más sencilla y fiable del tamaño de los paquetes que se transmiten, y hacer que los edificios no fueran un obstáculo absoluto, sino que en función del grosor y el material de los mismos pudieran comunicarse los vehículos en algunas circunstancias.

El principal problema que presenta el uso de OMNET++ en el desarrollo de las simulaciones es el tiempo de ejecución de las mismas. En [9], para la realización de alguna de las simulaciones podría estar ejecutándose semanas para obtener los resultados, mientras que con el uso de MASON el tiempo máximo de ejecución se reduce a un par de horas. Si se comparan los resultados de ambas simulaciones, se puede observar cómo las cifras no coinciden de forma exacta en todos los casos, pero la tendencia y las conclusiones que se pueden extraer son muy similares, lo que hace del uso de MASON una alternativa más atractiva para la continuación de este proyecto incluyendo comportamientos más complejos, pues la simulación basada en agentes dota de mucha más flexibilidad que un simulador como OMNET++.

5.2. Factores no tenidos en cuenta

Para la realización de este estudio se ha decidido simplificar el modelo con la intención de que en un periodo corto de tiempo pudiera ser realizado, obviando factores que pueden completarlo y que podrán ser desarrollados en un futuro, pero que en el momento de la implementación se decidieron dejar aparte. En este subapartado se expondrán los principales.

El primer factor que no se ha tenido en cuenta es la búsqueda por parte de los vehículos de más de un contenido. Durante el ciclo de vida de los vehículos en la simulación (que oscila en torno a los 300 segundos), los vehículos obtienen el contenido como máximo a los 120 segundos de entrar en el escenario, ya sea a través de la VANET, o porque lo han descargado por 3G, lo que hace que durante el resto del tiempo que circulan por la red solo comparten contenidos pero no se interesan por ninguno nuevo. Una mejora para una futura continuación del proyecto sería que al pasar un tiempo de haber conseguido un

contenido, intenten buscar otro nuevo, ya que en un entorno real los usuarios suelen acceder a varios contenidos.

Otro factor que suele aparecer en las redes actuales es la fragmentación de contenidos, ya que en el caso de descarga de vídeo, sonido o imágenes podrían no caber en una trama IP, lo que forzaría a dividir este contenido en varios trozos. Teniendo en cuenta que para las simulaciones se han supuesto los mensajes de contenidos de 1Kbyte, sería perfectamente asumible esta fragmentación al introducir contenidos de mayor tamaño. Esta adición mermaría las prestaciones de los protocolos que se han tenido en cuenta, en especial las de MHTI al transmitir a distancias mayores, en las cuales el movimiento de los vehículos puede producir más cortes de comunicación.

Un tercer factor que no se ha tenido en cuenta es el hecho de que no todos los vehículos pueden tener la voluntad de cooperar para que la comunicación total funcione, sino que pueden tener comportamientos egoístas (no ofrecen contenidos con la intención de ahorrar energía o ancho de banda), o maliciosos (los cuales pretenden obtener beneficios del resto de nodos). Para que en un sistema no sea atractiva la existencia de este tipo de nodos, sería necesario un sistema de incentivos que premiara la cooperación en la red y castigara a los nodos que no se comportaran del modo que deben. Este tema se trata en [2], y podría ser un objeto de estudio tras la realización de este trabajo.

6. Conclusions/Conclusiones

6.1. Versión en español

En este trabajo analizamos el funcionamiento de mecanismos de distribución de contenidos en redes vehiculares (VANET). La alta movilidad de los vehículos produce numerosas interferencias en las conexiones de una VANET, es por eso que el paradigma del sistema es el almacenamiento, transporte y reenvío, en el que los contenidos se pueden almacenar durante un tiempo en nodos antes de enviarlos hacia el destino. Esto es posible porque las aplicaciones relacionadas con la descarga de contenidos pueden admitir cierto retardo. Conforme a este esquema, se ha usado una simulación basada en agentes para estudiar tres protocolos (*Push*, *Pull* y MHTI; que son representativos de las distintas familias de protocolos propuestos en la literatura) para la distribución de contenidos en escenarios urbanos, utilizando la ciudad de Madrid como escenario de la simulación. En caso de que los vehículos no puedan conseguir el contenido usando la red vehicular, utilizarán una conexión auxiliar de 3G para garantizar que la experiencia del usuario no se vea afectada.

La mayor parte de los trabajos en la literatura relacionados con distribución de contenidos en VANET se centran en proponer un protocolo y estudian cómo se comporta (tasa de éxito, sobrecarga, retardo, etc.) sin tener en cuenta que la naturaleza de los contenidos cambia en función de cada aplicación. Siguiendo el trabajo realizado en [9], se ha estudiado cómo las características de las aplicaciones definidas en el número de contenidos a distribuir, y la popularidad de dichos contenidos impacta en las prestaciones de los distintos tipos de protocolos. También se ha considerado en el estudio la variación de la densidad de vehículos.

Coincidiendo con [9], en este trabajo se ha demostrado que la cantidad de contenidos tiene un gran impacto en la probabilidad de que un vehículo consiga obtener un contenido utilizando la VANET. Con un único contenido en la red, todos los protocolos muestran unas tasas de éxito elevadas, haciendo indistinguibles los unos de los otros observando este indicador. En cambio, al añadir al sistema una mayor variedad de contenidos, las prestaciones de los protocolos ya no son las mismas, mostrando unas tasas de éxito muy diferenciadas.

El protocolo *Push* muestra resultados aceptables con pocos contenidos (menos de 10), pero en el momento que este número aumenta, la tasa de éxito cae a 0, y la sobrecarga de la red tiende a infinito debido a que es casi imposible obtener el contenido deseado a través de la VANET. Ni siquiera vehículos que están cerca de una estación de información pueden obtener su contenido porque, aunque una estación de información tiene acceso a todos los contenidos, elige aleatoriamente el que transmite cada vez, por lo que es difícil que este contenido sea el que uno de sus vehículos vecinos está buscando.

La adición de popularidad al sistema evita la caída de la tasa de éxito a 0, sobre todo con tasas de popularidad superiores al 0.75, pero sin llegar a compensar la caída producida por el aumento de contenidos.

En el protocolo *Pull*, los vehículos cercanos a una estación de información consiguen su contenido incluso si hay cientos de contenidos (dado que lo piden proactivamente), es por eso que al aumentar contenidos la tasa de éxito se estabiliza en el 20%. Estos resultados tampoco son buenos, aunque la sobrecarga de la red es la más baja de los tres protocolos debido a que los contenidos solamente se transmiten bajo demanda, no como en el *Push*.

Este protocolo se ve más beneficiado que el anterior por la popularidad, alcanzando tasas similares a las de MHTI cuando alfa es 2 (una tasa de éxito cercana a 80% en el modo normal, y 85% con optimizaciones), pero estos valores de popularidad no son comunes en situaciones reales, por lo que no se podría presuponer que se hicieran efectivas.

El único protocolo capaz de ofrecer buenos resultados a partir de 100 contenidos es MHTI, el cual se estabiliza alrededor del 55% incluso con 100.000 contenidos. Esto significa que en más de la mitad de los vehículos, el protocolo encuentra una ruta para llegar a la estación de información sin producir una sobrecarga demasiado alta (sobre 16 bytes tienen que ser transmitidos para conseguir 1 byte de contenido, lo cual está muy alejado de la sobrecarga que producen los protocolos *Push*).

La existencia de popularidad hace mejorar aún más la tasa de éxito, mostrando tasas superiores al 70% en su versión optimizada a partir de un nivel de popularidad de 1, el cual no supone un nivel de popularidad poco realista.

Por otro lado, los contenidos almacenados en la memoria caché son un indicador de la diseminación de contenidos en la red. Los protocolos de tipo *Push* son los que tienen más contenidos almacenados, pero su comportamiento con respecto a la tasa de éxito es pobre, por lo que los protocolos de tipo *Push* encajan perfectamente con aplicaciones cuyo objetivo único sea diseminar contenidos, sin ningún vehículo objetivo. Por ejemplo, empresas podrían estar interesadas en este protocolo para distribuir anuncios publicitarios a través de la red, ofreciendo descuentos o productos especiales para el área en el que el anuncio se está diseminando. MHTI consigue mejores resultados cuando intenta conectar el vehículo con su objetivo (normalmente una estación de información), pero no almacena demasiados contenidos en caché.

Cuando se ha simulado la red vehicular usando el modelo actual de Internet, con una gran cantidad de contenidos (100000 en este caso) y un alfa de 0.8, se puede ver cómo *Push* y *Pull* no se ven beneficiados por el aumento de la densidad del tráfico, manteniendo sus bajas tasas de éxito, mientras que MHTI ve incrementada su tasa de éxito continuamente mientras aumenta el número de vehículos que circulan por las carreteras. Esto es un buen resultado, ya que con una densidad de tráfico baja, los vehículos pueden descargar contenidos usando conexiones 3G debido a que la red celular no está saturada, y en situaciones en las que el tráfico sea elevado, los vehículos podrán obtener el contenido por la VANET en la mayoría de los casos, lo que aliviará la congestión de la red celular.

El otro objetivo de este trabajo era comparar los resultados con los que se presentan en [9] para observar tendencias similares. Las conclusiones obtenidas en este proyecto son muy similares a las del trabajo anterior, el cual tenía en cuenta en sus simulaciones una representación fiel del nivel físico, con pérdidas de propagación en espacio libre, efecto de edificios en propagación, o colisiones de tráfico de datos producidas cuando dos o más vehículos transmiten al mismo tiempo. Estos parámetros hacen la simulación más realista pero necesitan muchos más recursos computacionales. En este trabajo se ha usado un modelo más simplificado, utilizando MASON y no OMNET++. El modelo de nivel físico es muy simplificado, éxito de comunicación solo cuando emisor y receptor están dentro de una distancia máxima y tienen visión directa. Sin embargo los resultados obtenidos son similares y reduce significativamente el tiempo de ejecución para simular la misma situación. MASON es un simulador basado en agentes, y una de sus características principales es la flexibilidad para estudiar comportamientos complejos y potencialmente distintos entre cada uno de los agentes, lo que hace este trabajo interesante con vistas a continuarlo y programar un sistema con comportamientos en los vehículos más complejos.

6.1.1. Líneas futuras de investigación

Una vez que se ha comprobado que MASON proporciona unos resultados similares a los realizados por un simulador específico de comunicaciones (OMNET++), el siguiente paso sería investigar cuál es el comportamiento de cada uno de los protocolos cuando los vehículos no solicitan un único contenido, sino que cada cierto tiempo vuelven a solicitar uno nuevo.

Por otro lado, sería interesante comprobar cuál es el comportamiento de la VANET cuando el contenido no puede ser enviado en un único mensaje, sino que debe ser dividido en varios “trozos” del tamaño que lo permita el protocolo de red. Esta asunción es muy probable, pues los contenidos tendrán un tamaño de 1Kbyte sino que será mayor, sobre todo en casos donde se estén descargando contenidos audiovisuales. El protocolo MHTI puede ser el más sensible a este cambio, ya que el realizar varios saltos para contactar con un nodo que tenga un contenido puede verse afectado por cambios en la topología de red cuando todavía no se ha finalizado la transmisión del contenido completo.

Por último, sería interesante aprovechar la capacidad del simulador basado en agentes para explotar la autonomía de la que gozan cada uno de los nodos, pudiendo simular situaciones en las que todos los agentes no mostraran el mismo comportamiento, sino que algunos decidieran no cooperar, ya sea por ahorrar recursos, o por producir un daño deliberado a la red. Sería necesario en ese caso establecer un mecanismo de incentivos para que un nodo que no esté dispuesto a cooperar no pueda beneficiarse de los servicios que ofrece la red, a través de premios a los nodos que cooperen o penalizaciones a los nodos que no lo hagan.

6.2. English version

In this work we analyze content distribution mechanisms in vehicular networks (VANET). The high mobility of vehicles produces lots of interferences in the VANET's connections, so the paradigm of the system is to store-carry-and-forward, where contents can be stored during a certain time in nodes before being sent to the target node. This is possible because applications based on contents download can deal with some delay. According to this scheme, we have used an agent-based simulator to study three protocols (Push, Pull and MHTI; each one represents one of the protocol families proposed in literature) for content distribution in urban scenarios, using the city of Madrid as evaluation scenario. If vehicles cannot get the content using the VANET after some time, they use a cellular backup connection to guarantee that the user experience is not affected.

Most works related to content distribution in VANET are focused in proposing a protocol and studying how it behaves (success rate, overhead, delay, etc.) but they seldom consider the nature of the contents consumed by the applications. Following the work in [9], we have studied how application characteristics, namely the number of contents to distribute and popularity of those contents, impact on each protocol performance. We have also studied the impact of traffic density on the performance.

Concurring with [9], in this work we have also proved that the amount of contents has a big impact in the probability that a vehicle has of getting a content using the VANET. With only one content in the network, every protocol shows a good success rate of getting the content through the VANET. However, with more contents in the network, each protocol has a different performance, so the number of contents affects different protocols in a different way.

The Push protocol shows acceptable results with few contents (less than 10), but when the number of contents increases its success rate falls to 0, and its overhead tends to infinity because vehicles hardly get any content through the VANET. Even vehicles that are close to an infostation can't get their contents because, although an infostation has access to all contents, the next content that an infostation broadcasts is chosen randomly, so it is unlikely that this content is the one the neighbor is searching.

The addition of popularity to the system allows the Push protocol to get a slightly better performance, especially with α higher than 0.75, but even with high α it doesn't compensate completely the low success rate produced by the number of contents.

In the one hop Pull protocol, vehicles near an infostation can get their contents even if there are thousands of contents (because they ask specifically for the one they want). That is why, when number of contents increases, success rate stabilizes at 20%. In any case, this is a poor performance, too, although overhead is the lowest among the three protocols because contents are only transmitted under demand, not as in Push.

The Pull protocol gets more benefit from popularity than Push, obtaining a performance similar to the one of the MHTI protocol when α is 2 (success rate around 80, and 85%

with optimizations), but such high popularity values are not common in real situations, so this performance is not realistic.

The only protocol capable of perform well with more than 100 contents is the MHTI protocol, which stabilizes to getting 55% of the contents through the VANET even with 100.000 contents. This means that in more than a half of the cases, the protocol finds a path to reach the closest infostation (perhaps after trying several times). Besides, it does so without producing an enormous overhead (around 16 bytes have to be transmitted to get one byte of content, which is lower than the overhead in Push protocols).

The existence of popularity makes the performance of the MHTI protocol even better, showing a success rate higher than 70% in the optimized version when popularity level is 1, which is not an unrealistic popularity level.

Another point to consider is that contents stored in cache are one indicator of content spreading over the network. Push protocols are the ones that have more contents stored, but their behavior with regard to success rate is poor, so push protocols fits perfect with applications whose purpose is only spreading contents, with no target vehicle. For instance, enterprises could be interested in this protocol to distribute advertisements through the vehicular network, offering discounts or special products in the area the advertisement is being distributed. MHTI performs better when it tries to connect at vehicle with a target (usually the infostation), but in this protocol the vehicles don't store many contents in their cache.

Finally, we wanted to study the behavior of the protocols with different traffic densities and when contents followed an Internet-like model, i.e. a big amount of contents (100000 in this case) and alpha 0.8. Both Push and one hop Pull protocols do not improve performance even increasing vehicle density, therefore keeping their poor performance. Instead, the MHTI success rate increases when the number of vehicles in the roads increases. This is a good result, because in scenarios with a low traffic density, vehicles can download contents using 3G taking advantage of the light data traffic in the cellular network, and in scenarios with high traffic density, vehicles mostly get their contents through the VANET, which mitigates cellular network overload.

Another purpose of this work was to compare our results with [9], which is a similar study using different simulation tools. The conclusions obtained in this project are pretty similar to the ones in [9], whose simulations consider detailed models for free space propagation losses, effects of buildings in signal propagation, and data traffic collisions produced when two or more nodes transmit at the same time. These parameters make the simulation more realistic but require lots of computer resources. In our work we have used a simpler model, using MASON and not OMNET++. Our physical layer model is quite simplified, a communication is successful only when emitter and receiver are close enough and they can see each other. However, the obtained results are similar and the execution time to simulate the same situation is significantly reduced. MASON is an agent based simulator, and one of his main features is the flexibility to study complex behaviors, potentially

different in each agent, which makes this work an interesting starting point to program a system with complex vehicles behaviors.

6.2.1. Future research

Once we have proved that MASON produces similar results to a specific communications simulator (OMNET++), the following step would be to find out how each protocol behaves when vehicles try to get more than one content in their trip, i.e. from time to time a vehicle is interested in a new content.

On the other hand, it could be interesting to explore how the VANET behaves when a content is not sent in one single message, but it is split into “chunks” whose size fits with the network protocol datagram capacity. This is a probable situation, because contents will typically have a size of more than 1Kbyte. The MHTI would be the most sensitive protocol to this change since some contents will be transmitted using multi-hop paths so, a topology change while transmitting the chunks of a content can result that two nodes stop seeing each other, so communication is not possible anymore and the content is not fully received.

Finally, it would be interesting to use the agent based simulator ability to model the agents’ autonomy, so simulations could include a heterogeneous distribution of nodes’ behaviors. For example, we can have a network in which some vehicles always cooperate to make the network work, but other vehicles have selfish or malicious behaviors, whose purpose is to save resources or harm the network. In this realistic situation, an incentive mechanism is needed to exclude from the network non cooperative nodes, rewarding nodes that show a good behavior, or punishing “bad” nodes. The simulation framework developed in this project (SUMO+MASON) seems a good tool to study this kind of scenario.

7. Bibliografía

- [1] M. Conti y S. Giordano, «Mobile Ad Hoc Networking: Milestones, Challenges, and New Research Directions,» *IEEE Communications Magazine*, vol. 52, nº 1, pp. 85-96, 2014.
- [2] J. A. Dias, J. J. Rodrigues y L. Zhou, «Cooperation advances on vehicular communications: A survey,» *Vehicular Communications*, vol. 1, pp. 22-32, 2014.
- [3] H. Moustafa, «Introduction to Vehicular Networks,» de *Vehicular networks: techniques, standards, and applications*, CRC Press, 2009.
- [4] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin y T. Weil, «Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions,» *IEEE communications surveys & tutorials*, vol. 13, nº 4, pp. 584-616, 2011.
- [5] ETSITR102638, «Intelligent Transport System (ITS); Vehicular Communications; Basic Set of Applications; Definition,» ETSI Std. ETSI ITS Specification TR 102 638 version 1.1.1, 2009.
- [6] C2C-CC, «Car to Car Communication Consortium Manifesto: Overview of the C2C-CC System,» *Car to Car Communication Consortium*, nº Version 1.1, 2007.
- [7] W. -. W. L. W. Group y L. S. Committee, «802.11p-IEEE Standard for Information technology- Local and metropolitan area networks- Specific requirements-- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6:Wireless Access in Vehicular Environments,» 2010.
- [8] J. Kurose y K. Ross, «Protocol layers and their service models,» de *Computer networking: a top-down approach*, Pearson, 2013, p. 51.
- [9] M. Urueña, I. Soto, I. Martinez-Yelmo y M. Calderon, «Effect of content popularity and number of contents on the performance of content distribution protocols in urban VANET scenarios,» *Paper submitted for publication and under review*, 2015.
- [10] B. Xu, A. Ouksel y O. Wolfson, «Opportunistic Resource Exchange in Inter-vehicle Ad-hoc Networks,» *IEEE International Conference on Mobile Data Management*, pp. 4-12, 2004.
- [11] L. Wischhof y H. Rohling, «Information Dissemination in Self-Organizing Intervehicle Networks,» *IEEE transactions on intelligent transportation systems*, vol. 6, nº 1, pp. 90-101, 2005.

- [12] J. Zhao, Y. Zhang y G. Cao, «Data Pouring and Buffering on the Road: A New Data Dissemination Paradigm for Vehicular Ad Hoc Networks,» *IEEE Transactions on Vehicular Technology*, vol. 56, nº 6, pp. 3266-3277, 2007.
- [13] Y. Lim, S. Ahn y K.-H. Cho, «Abiding Geocast for Commercial Ad Dissemination in the Vehicular Ad Hoc Network,» *IEEE International Conference on Consumer Electronics (ICCE)*, pp. 115-116, 2011.
- [14] C. Maihöfer, T. Leinmüller y E. Schoch, «Abiding Geocast: Time-stable Geocast for Ad Hoc Networks,» *2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2005)*, pp. 20-29, 2005.
- [15] H. Wu, R. Fujimoto, R. Guensler y M. Hunter, «MDDV: A Mobility-Centric Data Dissemination Algorithm for Vehicular Networks,» *1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, pp. 47-56, 2004.
- [16] D. Sormani, G. Turconi, P. Costa, D. Frey, M. Migliavacca y L. Mottola, «Towards Lightweight Information Dissemination in Inter-Vehicular Networks,» *3rd International Workshop on Vehicular Ad Hoc Networks (VANET 2006)*, pp. 20-29, 2006.
- [17] I. Leontiadis, P. Costa y C. Mascolo, «Persistent Content-based Information Dissemination in Hybrid Vehicular Networks,» *IEEE International Conference on Pervasive Computing and Communications (PerCom 2009)*, pp. 1-10, 2009.
- [18] M. Amadeo, C. Campolo y A. Molinaro, «CRoWN: Content-Centric Networking in Vehicular Ad Hoc Networks,» *IEEE communications letters*, vol. 16, nº 9, pp. 1380-1383, 2012.
- [19] I. Lequerica, P. M. Ruiz y V. Cabrera, «Improvement of Vehicular Communications by Using 3G Capabilities to Disseminate Control Information,» *IEEE Network*, nº 24, pp. 32-38, 2010.
- [20] G. Rémy, S.-M. Senouci, F. Jan y Y. Gourhant, «LTE4V2X - Impact of High Mobility in Highway Scenarios,» *Global Information Infrastructure Symposium (GIIS 2011)*, pp. 1-7, 2011.
- [21] C. J. Bernardos, I. Soto, M. Calderón, F. Boavida y A. Azcorra, «VARON: Vehicular Ad-hoc Route Optimisation for NEMO,» *Computer Communications*, nº 30, pp. 1765-1784, 2007.
- [22] M. Gramaglia, C. J. Bernardos y M. Calderón, «Seamless internet 3G and opportunistic WLAN vehicular connectivity,» *EURASIP Journal on Wireless Communications and Networking*, pp. 1-20, 2011.

- [23] C. M. Macal y M. J. North, «Tutorial on agent-based modelling and simulation,» *Journal of Simulation*, vol. 4, n° 3, pp. 151-162, 2010.
- [24] A. Borshchev y A. Filippov, «From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools,» *The 22nd International Conference of the System Dynamics Society*, 2004.
- [25] «SUMO,» [En línea]. Available: http://sumo.dlr.de/wiki/Main_Page. [Último acceso: Junio 2015].
- [26] «MASON Multiagent Simulation Toolkit,» [En línea]. Available: <https://cs.gmu.edu/~eclab/projects/mason/>. [Último acceso: Junio 2015].
- [27] C. Fricker, P. Robert, J. Roberts y N. Sbihi, «Impact of traffic mix on caching performance in a content-centric network,» *2012 Proceedings IEEE INFOCOM Workshops*, pp. 310 - 315, 2012.
- [28] G. Kotusevski y K. A. Hawick, *Research Letters in the Information and Mathematical Sciences*, vol. 13, pp. 35-54, 2009.

Anexo I. Acrónimos y siglas

C2C-CC (Car 2 Car Communication Consortium)

CALM (Continuous Air-interface Long and Medium range)

CEN (European Committee for Standardization)

CENELEC (European Committee for Electro-technical Standardization)

DSRC (Dedicated Short-Range Communications)

DTN (Delay Tolerant Networks)

ETSI (European Telecommunications Standards Institute)

IEC (International Electro-technical Commission)

IEEE (Institute of Electric and Electronic Engineers)

IHVS (Intelligent Vehicle Highway Systems)

ISO (International Organization for Standardization)

ITS (Intelligent Transportation System)

ITU (International Telecommunication Union)

MANET (Mobile Ad-Hoc NETwork)

MASON (Multi-Agent Simulation of Networks)

MHTI (Multi-Hop to Infostation)

SAE (Society of Automotive Engineers)

SiMTraM (Simulation of Mixed Traffic Mobility)

SUMO (Simulation of Urban Mobility)

TraaS (TraCI as a Service)

TRACI (TRAffic Control Interface).

V2I (Vehicle 2 Infostation)

VANET (Vehicular Ad-Hoc Networks)

VDTN (Vehicular Delay Tolerant Networks)

V2V (Vehicle 2 Vehicle)

WAVE (Wireless Access in Vehicular Environments)

Anexo II. Planificación y costes

Planificación

Uno de los aspectos clave a la hora de realizar un proyecto es la planificación, la cual permite trazar unos objetivos para conseguir en un tiempo limitado, aumentando así la eficiencia para desarrollar el trabajo, y además tener una referencia sobre si una actividad se está retrasando y el impacto que puede tener esto sobre el desarrollo del proyecto.

El proyecto tiene una duración total de 120 días, repartidos en los meses de noviembre de 2014 hasta mayo de 2015, en los cuales se han tenido en cuenta las festividades de Navidad y semana santa en las cuales no se realiza el trabajo, además de los fines de semana. Cada jornada de trabajo está formada por 3 horas, lo que hace un total de 360 horas dedicadas al proyecto, incluyendo la elaboración de la memoria.

A continuación se expondrán las distintas actividades que se han planificado con una breve explicación de qué subtareas se llevan a cabo y el tiempo estimado para realizarlas:

- A. Documentación y lectura sobre redes vehiculares y protocolos de distribución de contenidos. Tras la elección del proyecto, la primera acción a realizar debe ser la búsqueda de información sobre los temas que se tratarán, ya que no se puede desarrollar un entorno de simulación como el que se va a realizar sin unos sólidos conocimientos y unos conceptos claros. Además de la lectura sobre redes vehiculares, también es necesaria la adquisición de conocimientos sobre qué es la simulación basada en agentes, la cual supone un paradigma nuevo de programación, y tiene unas características muy concretas. También es necesario profundizar en la lectura sobre los protocolos que se utilizarán en la simulación, ya que son el pilar fundamental del código de programación. (21 horas, 7 días)
- B. Instalación del entorno de trabajo. Una vez comprendido el problema que se va a tratar, el siguiente paso es preparar y dejar listo todo el entorno que será requerido durante el desarrollo, esto es: Instalación de programas y paquetes en Ubuntu (que será el sistema operativo utilizado), y la realización de pequeñas pruebas en SUMO y MASON para comprender su funcionamiento y familiarizarse con sus modos de uso, centrándose en las opciones que serán útiles para el proyecto que se va a desarrollar. (15 horas, 5 días)
- C. Establecimiento de la comunicación entre ambos programas. Una vez instalado y probado todo el entorno, es necesario revisar las opciones que se tienen para comunicar los dos simuladores, e implementar la más idónea. Es necesario realizar pequeños programas de prueba que permitan al desarrollador entender todas las opciones que ofrece la herramienta elegida (en este caso, TRACI) y aprender a utilizar las funciones que serán de utilidad para el desarrollo del proyecto. (18 horas, 6 días)

- D. Visualización de mapas y de agentes en MASON. La visualización del estado de la simulación resulta esencial para comprobar la efectividad de la situación que se está simulando. Es necesario comprobar que el movimiento de cada uno de los agentes corresponde con el movimiento de vehículos que genera SUMO, o si los edificios colocados en MASON coinciden con los edificios que existen en el plano original que se ha tomado. MASON ofrece ciertas dificultades para cargar información de gráficos, por lo que también será necesario instalar librerías extra para poder realizar la visualización de los edificios. (54 horas, 18 días)
- E. Programación. Implementación de los distintos protocolos y sistema general. En esta etapa se realiza el grueso del proyecto. Es necesario establecer las especificaciones exactas que tendrán cada uno de los protocolos además de las características comunes, como tiempos de transmisión, número de vehículos, distancia mínima de las rutas que podrán tomar los mismos, cuáles son las prestaciones que se considerarán como aceptables para dar por válido un bloque ya programado. En cuanto a desarrollo del código, será necesario crear una pequeña interfaz que permita a un usuario que desee realizar una simulación introducir los parámetros que desea medir, y hacer un código claro y limpio que pueda ser utilizado por otras personas en momentos posteriores. Esta actividad quedará finalizada cuando se hayan implementado todos los protocolos de difusión de contenidos. (75 horas, 25 días)
- F. Realización de pruebas y modificaciones del código. Una vez el código está desarrollado, es necesario realizar pruebas sobre cada uno de los protocolos para comprobar si durante la ejecución de la simulación se lanzan mensajes de error que interrumpan la actividad, si los resultados son coherentes y, en caso de ser necesario, realizar los cambios oportunos en el código para subsanar estos errores. El periodo de pruebas es el más largo de todos, pues es el que requiere una mayor supervisión pues una vez se dé por finalizado el mismo, el entorno de simulación se dará por bueno. (102 horas, 32 días)
- G. Elaboración de la memoria. Una vez finalizado el desarrollo del simulador y se hayan recopilado todos los resultados, es necesario registrar y documentar todo el trabajo realizado para su posterior presentación y valoración. Esta última actividad debe ser realizada facilitando la comprensión y la lectura del documento, ya que una vez entregada puede ser útil para la realización de proyectos similares, o incluso una continuación del trabajo hacia vías que no se han contemplado como las propuestas en el apartado de alternativas de diseño. (81 horas, 27 días)

Las actividades que se han propuesto anteriormente están representadas en el tiempo a través de un diagrama de Gantt en la Ilustración 0-1, en la cual se puede apreciar la duración estipulada para cada actividad. Se ha definido que al día siguiente de acabar de realizar una actividad se empieza la siguiente, por lo que no se solapan en el tiempo.

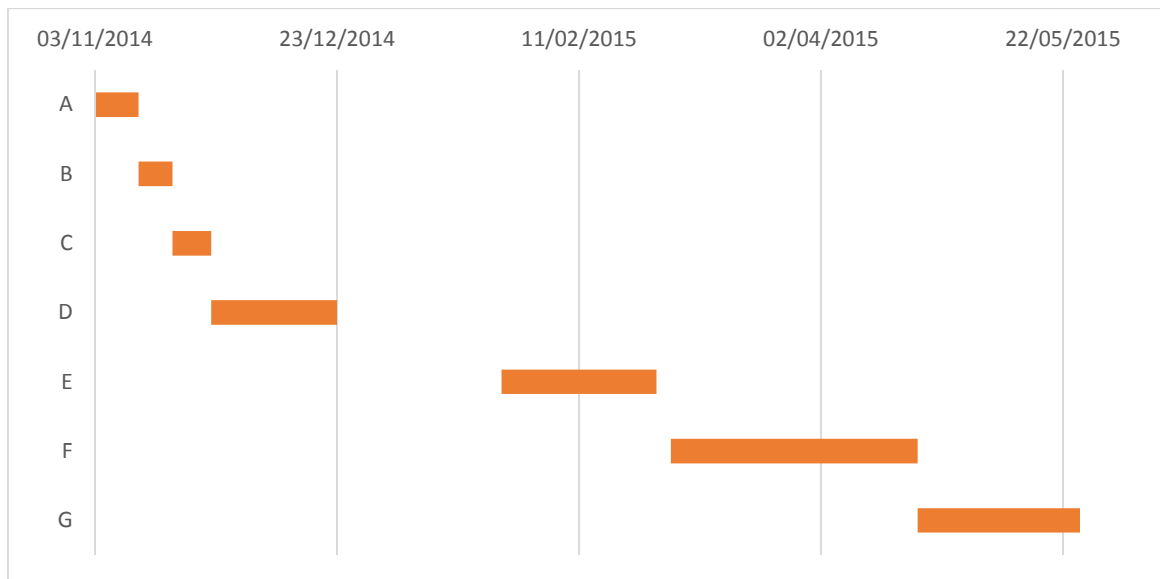


Ilustración 0-1. Diagrama de Gantt

Desde el 23 de diciembre hasta febrero no hay actividad programada pues corresponde al periodo vacacional de Navidades, y al período de exámenes de primer cuatrimestre, por lo que no está previsto realizar ningún avance durante esas fechas. En la actividad F durante la semana santa tampoco se realizará actividad, aunque esto no aparezca reflejado en el diagrama.

Costes y presupuesto

Materiales necesarios

El material esencial para la realización del proyecto es un ordenador de sobremesa con un procesador y una memoria RAM adecuada para la función que desempeñará, ya que simular el tráfico en una ciudad con miles de vehículos que además se comunican entre ellos requiere un coste computacional elevado, por lo que un ordenador de prestaciones no muy altas puede suponer un aumento del tiempo de simulación importante. Es por eso que el ordenador que se ha utilizado para el desarrollo ha sido un GLite con un procesador de 8 núcleos Intel Core i7-3770 con arquitectura de 64 bits con una memoria RAM de 16 GB y un disco duro de almacenamiento de 1 TB. Se estima que el ciclo de vida de este tipo de dispositivo es de 4 años, dato que se utilizará para la amortización.

Para la utilización del ordenador de sobremesa han sido necesarios algunos periféricos que permiten la interacción con el mismo, como son un teclado, un ratón y una pantalla. Todos estos dispositivos se han considerado con un ciclo de vida igual al del ordenador de sobremesa.

El sistema operativo utilizado en dicho ordenador ha sido Ubuntu 14.04, el cual tiene una licencia gratuita al pertenecer al grupo de sistemas operativos Linux. Además, todo el software perteneciente al entorno de simulación es *open source* por lo que no ha tenido coste alguno su utilización.

Sí ha sido necesario el uso del paquete de ofimática Office 2013 para la creación de tablas y gráficas para visualizar los resultados de las simulaciones, y para la consiguiente generación de esta memoria donde se refleja el trabajo realizado. Para su utilización se ha utilizado una licencia anual de carácter básico.

Gastos de personal

En la realización del proyecto han contribuido dos ingenieros: El primero, el estudiante del grado en Ingeniería en Tecnologías de Telecomunicación, el cual ha trabajado un total de 360 horas en el proyecto. Se ha considerado su título profesional equiparable a un ingeniero junior, por lo que se ha estimado el salario por hora en 20 euros.

Por otro lado, un ingeniero senior se ha encargado de asesorar al estudiante en el desarrollo del proyecto a través de un total de 30 horas de tutorías en las que se han resuelto dudas y se han comentado los resultados para estudiar si podrían tener algún tipo de error o no. El salario estimado por hora para este nivel de cualificación se ha estimado en 33 euros.

Costes indirectos

Existen una serie de gastos que no se pueden contabilizar directamente como el gasto de electricidad para alimentar los equipos o el coste de la conexión a Internet para las consultas que se han tenido que realizar, por lo que se ha supuesto que éstos costes supondrán un 20% de los costes anteriormente considerados como aparece en el presupuesto (página 75).

Presupuesto detallado

En la siguiente página se expone un presupuesto donde se detallan cada uno de los gastos que se han mencionado en apartados anteriores. Todas las cantidades incluyen un 21% de IVA (Impuesto sobre el Valor Añadido), el cual es el que está vigente en la legislación actual española.



UNIVERSIDAD CARLOS III DE MADRID
Escuela Politécnica Superior

PRESUPUESTO DEL PROYECTO

1.- Autor: Juan José Campaña Montes

2.- Departamento: Ingeniería Telemática

3.- Descripción del Proyecto:

- Título: Estudio de mecanismos de distribución de contenidos en redes vehiculares urbanas
- Duración (meses): 6
Tasa de costes Indirectos: 20%

4.- Presupuesto total del Proyecto (valores en Euros):

9.941,40 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	Categoría	Dedicación (horas)	Coste persona hora	Coste (Euro)
Ignacio Soto Campos	Ingeniero Senior	30	33,00	990,00
Juan José Campaña Montes	Ingeniero	360	20,00	7.200,00
			Total	8190

MATERIALES

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ⁴⁾
Ordenador	600,00	100	6	48	75,00
Periféricos	20,00	100	6	60	2,00
Licencia Office	70,00	50	6	12	17,50
				Total	94,5

⁴⁾ Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado
B = periodo de depreciación (60 meses)
C = coste del equipo (sin IVA)
D = % del uso que se dedica al proyecto (habitualmente 100%)

6.- Resumen de costes

Costes Totales	Total
Personal	8.190
Amortización	95
Costes Indirectos	1.657
Total	9.941

Anexo III. Manual de instalación del entorno de simulación

A continuación, presentaremos un breve manual de instalación del simulador de agentes MASON junto con el simulador de tráfico SUMO para la recreación de la comunicación en una red VANET entre vehículos en un entorno urbano.

SUMO

SUMO (Simulation of Urban MObility) es un simulador de tráfico urbano, que a partir de un mapa en formato xml, puede simular el movimiento de vehículos a través de ese mapa. SUMO será el encargado de generar el movimiento de los vehículos, y ésta información será enviada para su utilización.

Desde la página principal de SUMO se pueden descargar los archivos binarios para su instalación:

http://sumo.dlr.de/wiki/Main_Page

Además, las librerías libfox, libgdal y libxerces son necesarias para su uso, las cuales se pueden obtener de los repositorios de Ubuntu:

```
sudo apt-get install libfox-1.6-dev libgl1-mesa-dev libglu1-mesa-dev
```

```
sudo apt-get install libgdal1-dev proj libxerces-c2-dev
```

```
sudo apt-get install libxerces-c-dev
```

```
sudo ln -s /usr/lib/libgdal1.7.0.so /usr/lib/libgdal.so
```

Cuando esté todo instalado, se puede proceder a ejecutar SUMO desde la consola a través del comando sumo o sumo-gui.

Comandos consola generación de vehículos para la simulación en SUMO

Para la realización del experimento, SUMO necesita un fichero de rutas donde podrá leer los vehículos que va insertando en la simulación, su origen y su destino. El archivo de rutas puede generarse manualmente, decidiendo en qué instantes entran los vehículos a la simulación, y cuál será su origen y destino. Para simulaciones con pocos vehículos esta opción puede ser interesante, pero en el caso que nos ocupa, con miles de vehículos, usaremos una herramienta de generación automática de rutas que nos ofrece SUMO. Con los siguientes comandos se generan primero unas rutas aleatorias y luego redistribuye el tráfico para que todos los vehículos no tiendan a circular por las vías principales, lo que produciría numerosos atascos.

```
.../sumo-0.23.0/tools/trip/randomTrips.py -n madrid.net.xml -l -p 1.8 -o 2000_1_trips.trips.xml --min-distance=1000
```

```
../sumo-0.23.0/tools/assign/dualterate.py -n madrid.net.xml -t 2000_1_trips.trips.xml -l 30 -C
```

El parámetro `-p` indica cada cuánto se insertarán vehículos en la simulación, en este caso 1.8 para generar 2000 vehículos. El parámetro `-o` indica el nombre del archivo de salida que se generará, es decir, el archivo en el que aparece el momento de aparición de cada vehículo y la ruta que seguirá. Por último, se ha utilizado el parámetro `--min-distance` para indicar que las rutas no pueden ser inferiores a 1 km, para así evitar trayectos excesivamente cortos, los cuales no favorecen la comunicación en una red VANET.

MASON

MASON (Multi-Agent Simulator Of Networks), es un simulador de agentes diseñado en el lenguaje Java, el cual permite realizar y visualizar las simulaciones que se programen. Desde la página principal de MASON se puede descargar el mismo:

<http://cs.gmu.edu/~eclab/projects/mason/>

Para que un proyecto java utilice MASON es necesario copiar la carpeta `sim` y la carpeta `ec` dentro del directorio `src` del proyecto en el que se desee utilizar (asumimos el uso de la herramienta Eclipse para llevar a cabo la simulación, de ahí la estructura de carpetas.).

Para el correcto funcionamiento del simulador (visualización 3D, utilización de algunas clases) son necesarias algunas librerías extra que te ofrecen los propios desarrolladores de MASON en su página. Los archivos `.jar` deberán ser añadidos al `classpath` del proyecto para que MASON pueda funcionar.

El listado de todos los archivos añadidos al `classpath` es el siguiente

`commons-math3-3.4.1.jar`

`gdal-1.8.1.jar`

`geomason.1.5.jar`

`gt-data-11.0.jar`

`itext-1.2.jar`

`j3dcore.jar`

`j3dutils.jar`

`jcommon-1.0.21.jar`

`jdom.jar`

jfreechart-1.0.17.jar

jmf.jar

JTS_Test.jar

jts-1.13.jar

jtsio-1.13.jar

junit.jar

log4j-over-slf5j-1.6.4.jar

log4j.jar

mason.18.jar

org.harmcrest.core_1.1.0.v20090501071000.jar

portfolio.jar

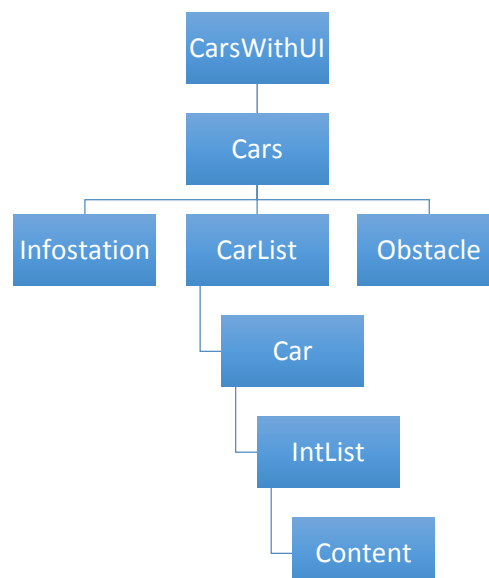
vecmath.jar

xerces.jar

xercesImpl.jar

Estos archivos se pueden encontrar en la carpeta lib de SUMO o TRACI4J, en la carpeta libraries de MASON, en la carpeta lib de jts o en el paquete de geotools.

ESTRUCTURA DE CLASES



Como se puede observar en el organigrama superior, la clase principal es CarsWithUI, la cual contiene un método main que ejecuta todo el código. CarsWithUI implementa la interfaz gráfica del simulador, mostrando los agentes (Infostations y Vehículos) y los edificios.

CarsWithUI crea un objeto Cars, que es el encargado de realizar la simulación pues inicializa el escenario donde se moverán los agentes, y gestiona el movimiento y la interacción de los mismos. La clase Cars incluye también un método main que ejecuta el código sin interfaz gráfica para optimizar el tiempo de ejecución.

En Cars se instancian cuatro Infostations (agentes que representan las estaciones con conexión a Internet, y que proveen de contenidos a los vehículos), un CarList, que es la lista de vehículos (Car) que aparecen en la simulación, y todos los obstáculos (edificios) que habrá en la simulación, utilizando la clase Obstacle para representarlos. Asimismo, para la correcta utilización del protocolo Push optimizado se usará una lista que almacena los contenidos (clase Content), los cuales contienen el identificador del contenido y su marca de tiempo para su correcta elección a la hora de transmitir. La gestión de este tipo de contenidos (que son diferentes por la marca de tiempo) se realiza a través de la clase IntList.

OBSTÁCULOS

Para la correcta carga y visualización del mapa de obstáculos (edificios), desde la clase Cars se utiliza el método loadxml para cargar un archivo de extensión .xml que incluya los obstáculos de la ciudad en la que se desea realizar la simulación. Si se desea cambiar el archivo de obstáculos, es necesario reescribir la ruta del fichero en la siguiente línea de código:

```
File file=new File("madrid/madrid.poly.xml");
```

GEOMASON

Para la simulación que se está llevando a cabo, se exige la utilización de geometrías para el tratamiento de los edificios y su visualización. La herramienta que usaremos para solucionar este problema es Geomason.

Geomason es una extensión de MASON para el uso de figuras geométricas y formas. Es necesario para trabajar con los mapas en el mismo debido a las limitaciones que ofrece el simulador de agentes por defecto.

<http://cs.gmu.edu/~eclab/projects/mason/extensions/geomason/>

También se utilizan algunas librerías de Geotools.

<http://www.geotools.org/>

TRACI4J

Durante la simulación, se necesita comunicar los dos programas (MASON Y SUMO) para saber cuál es la posición de cada uno de los vehículos en cada paso de la simulación. La herramienta elegida para realizar esta tarea es Traci4J, cuyo código fuente se puede descargar desde el siguiente enlace:

<https://github.com/egueli/TraCI4J>

Traci4J es una librería que permite comunicar cualquier programa Java con SUMO. Tras descargar los archivos del enlace superior, es necesario incluir los directorios de e it que se encuentran dentro de src/java, en el directorio src del proyecto java que se vaya a utilizar.

VISUALIZACIÓN DE LA SIMULACIÓN

VISUALIZACIÓN DE SUMO

```
private static final String SIM_CONFIG_LOCATION =  
    "madrid/madrid.sumo.cfg";
```

En SIM_CONFIG_LOCATION se escribe la ruta en la que se encuentra el archivo de configuración de la simulación que se quiere realizar.

```
System.setProperty("it.polito.appeal.traci.sumo_exe",  
    "/home/juanjo/Documentos/Vehicular_networks/SUMO/sumo-  
0.22.0/bin/sumo");
```

La instrucción anterior debe aparecer en el código si no se desea que aparezca la interfaz gráfica de SUMO.

VISUALIZACIÓN DE MASON

Para que aparezca la consola de MASON para parar la simulación, finalizarla u obtener información de los agentes es necesario que en el main de CarsWithUI aparezcan estas líneas de código:

```
Console c = new Console(vid);  
c.setVisible(true);
```

Además, en el método updateCar de la clase Cars deben aparecer las siguientes instrucciones:

```
yard.setObjectLocation(car,new Double2D(xpos, ypos));  
schedule.scheduleRepeating(car);
```

Para realizar las simulaciones de la forma más eficiente en tiempo, sólo será necesario ejecutar el programa eliminando las líneas de código que se han comentado anteriormente, además de ejecutar el main de Cars y no el de CarsWithUI.

VALORES DE LA SIMULACIÓN

Al ejecutar el main (ya sea desde CarsWithUI o Cars) aparecerá un menú por la consola para seleccionar el protocolo que se desea simular. Al seleccionar alguno, después se pedirá el número de contenidos distintos que se asumen, y el parámetro alpha de la distribución Zipf, por el cual un valor de 0 supondrá una probabilidad uniforme de que se requiera cualquier contenido. Un valor mayor supondrá la existencia de popularidad entre los contenidos, por lo cual algunos tendrán más probabilidad de ser requeridos que otros.

En el menú principal, aparte de aparecer todos los protocolos, también se da la opción de simular todos los protocolos, con un número de contenidos que empezará por 1 e irá multiplicándose por 10, y un valor de alpha desde 0 a 2 en saltos de 0.25. Para modificar estos parámetros de la simulación, es necesario encontrar las siguientes líneas de código:

```
for(alpha=0;alpha<2.1;alpha=alpha+0.25){  
  
for(contents=1;contents<1000000;contents=contents*10){  
  
for(protocol=1;protocol<7;protocol++){
```

VISUALIZACIÓN DE RESULTADOS

Los resultados de cada una de las simulaciones se guardarán en un archivo de texto plano separados por un tabulado para su correcta conversión a una tabla Excel. La ruta se deberá modificar en la clase CarList dentro del método listpoints, el cual se encarga de recuperar los datos que nos interesan de los vehículos, como las retransmisiones que han hecho, o si han conseguido el contenido:

```
fw=new FileWriter("/home/juanjo/Documentos/simulations",true);
```

Appendix IV. Extended abstract

Our society is demanding year by year more investment in new technologies and more products to make real the concept “Internet everywhere”. Laptops or smartphones are not enough to satisfy consumer needs, so everyday new objects are being improved to offer connectivity based functionality they couldn’t offer before. One example is the rising market of wearables (smartwatches, rings, collars...). The next logical step is to give this connectivity to objects where people spend hours traveling, namely vehicles.

The aim of this work is to study different mechanisms to distribute contents in vehicular networks so each vehicle can get its desired content in a certain time (if this deadline is exceeded, vehicles will download their content using a cellular network). This study has been carried out using an agent based simulation model, a flexible model to simulate systems based on complex interactions among autonomous entities. So, another purpose of this work has been to test a simulation environment that could be the basis for future simulation studies in the field of vehicular networks, in particular when more complex and heterogeneous behaviors of the vehicles are present.

The first option to connect vehicles is to use a cellular network as it has been used for previous devices as smartphones or tablets. Since data traffic generation by the users is increasing exponentially, ISPs have difficulties to deal with a new source of traffic, especially if this source is generating a big amount of traffic, as it could be expected when users have the chance to download contents when they are in their cars. Furthermore, there are obstacles in cities like tunnels or low connectivity areas where connection is not possible. Fortunately, we have an alternative, the possibility of connecting vehicles to each other via a short range connection.

This kind of networks, based on connecting mobile nodes, has been already developed for nodes as laptops, emergency rescue teams, etc. These networks are called MANETs. The problem here is that this networks can’t deal with high mobility nodes as vehicles, because their protocols are adapted to route traffic around paths that are static during a certain time, but this condition is not met when cars are moving through streets with a big amount of obstacles (buildings). Therefore, the communication paradigm must change.

That is why vehicular networks solutions were developed, to create a specific network that could address the problem of high mobility and intermittent communications. There are three main types of vehicular networks: VANETs, DTNs and VDTNs. VANETS are the direct evolution of MANETs, where old protocols were adapted to make possible a connection between two nodes. When a node wants to establish a communication, a path is created using the network nodes at any given time. VANETs work worse in situations with high level of interruptions and disconnections. Next, DTNs implement the store-carry-and-forward paradigm, where vehicles transport information in their cache memory and they forward it when they found another node interested in the information or that can reach other nodes interested in the information. Finally, VDTNs are one evolution of DTNs whose main difference is that the VDTN adds a control plane to save bandwidth

making content transmissions only when there is a good possibility that they are going to be successfully.

Even though there are three types of vehicular networks, all of them are denoted as VANET in the literature, so in this work we will use this term to refer to any kind of vehicular network.

Vehicular networks can be used for many different applications, which may be classified in three categories: safety, traffic efficiency and infotainment. Safety applications were the first ones designed to be used in vehicular networks; in fact they were the original reason for the creation of vehicular networks. Connecting vehicles to each other could save many lives advising emergency services in real time when an accident occurs. It also can be used to warn drivers when they are going to change their lane to overtake another vehicle, or when they are driving in the wrong direction. Accident prevention could save a lot of money and it could avoid a lot of human damage.

Another type of application is traffic efficiency, where vehicles could cooperate to distribute traffic among different roads and prevent traffic congestion, or vehicles can assist the driver about the speed she should take to avoid unnecessary stopping.

The last type of application is infotainment. Vehicles have not power limitations, so people can use them to download contents from the Internet (like media streaming, or web navigation) or they can use vehicles to check traffic conditions or prices of nearby shops as gas stations or food stores. Infotainment applications are studied in this work. Because of their features, like the size of contents, and that they can deal with some delay, DTNs fit perfectly with these applications.

Although VANETs are a new concept of technology and it is not implemented yet in most countries, there has been many efforts to create standards related to this field, and organizations as IEEE (whose protocol IEEE 802.11 is the one used for wireless local area network transmissions, and has created the 802.11p protocol, specific for vehicular communications), ISO, CEN or ETSI have been working to define requirements, specifications or what is expected about this technology. Moreover, Japan is the more advanced country in VANET because in 2008 around 20 million of vehicles were equipped with ETC OBUs, technology related to vehicular networks.

As it was already said, content distribution is not critic in time, because one person who request one content can wait a few seconds to get it while he is driving, or sitting in the car. Nevertheless, the content must be provided to the car before it loses its interest. In vehicular networks we have also specialized nodes called infostation, which is a fixed node connected to the VANET and located ideally in main intersection roads where most vehicles have to pass through. Infostations have access to Internet, so they can send any content requested by vehicles.

Many different protocols have been proposed in the literature to carry out content distribution, but they can mainly be grouped in two families: Push protocols, and Pull

protocols. In Push protocols, contents are broadcast from time to time, so nearby vehicles (neighbors) can receive this content. If one vehicle receives a content and it is interested on it, he uses it immediately. In Pull protocols vehicles only send a content on demand, i.e. when a vehicle is interested in a particular content, it sends a request message to all of its neighbors and if some of them has the content, it replies with content. Also it should be mentioned that there are multi-hop Pull protocols in which a request is not only sent to neighbors, but to vehicles that can't be directly reached at one hop, but can be reached by means of multiple hops through nearby cars that retransmit the request until it gets its target.

Most protocols in the literature reviewed in this work proposes scenarios with only one content or at least a small number of contents, where it is easy for vehicles to get their desired content using the VANET, because every vehicle is asking and storing that same content. The existence of popularity among contents has also been usually ignored when evaluating protocols for content distribution in VANET. But this is a useful parameter because in real environments people don't demand every content with the same probability. A final important consideration is the need for a 3G/4G network as a backup when it is not possible to get a content through the VANET. This guarantees the service even if the VANET is not able to provide the content. Again, many of the reviewed protocols in the literature do not implement a 3G/4G network as backup. In our work, as we will describe, we consider a 3G network backup, many contents, and popularity.

In this work we study the performance of three reference protocols: a Push protocol, a Pull protocol and the MHTI protocol (a multi-hop content distribution protocol proposed in [9]). We analyze the performance in different situations (different traffic density, different number of contents, different popularity levels) and compare results to [9]. Although this work is similar to [9], the main difference between them is the simulation platform. In [9] the authors use OMNET++, a specific simulator for communications, in which it is not easy to introduce some kind of modifications, specifically those related with the behavior of the vehicles. Additionally, it considers in a very realistic way physical parameters as propagation losses and data traffic collisions when two vehicles transmit at the same time, so it consumes a lot of computational resources.

We have created a simulation platform to evaluate content distribution using SUMO to generate the movements of vehicles through the city of Madrid, and MASON (replacing the role of OMNET++ in [9]) to get their positions and carry out communications between vehicles. SUMO is an open source tool that can generate vehicular traffic in a city using a map in xml format and a file with routes (that can be generated automatically). SUMO also provides a tool to communicate with him from an external program to get simulation information. This tool is called TRACI. As MASON is developed in Java, we have used TRACI4J (library for Java that implements TRACI's interface) to ask for vehicles position in each step of the simulation.

MASON is an agent based simulator that takes the vehicles coordinates and creates agents that perform the communications. In an agent based model we have agents entities that

are autonomous, i.e. their movements or actions are decided by themselves (their own code, their own state variables). Agents interact with each other and with the scenario in which they are moving, so they can change his state. Following this model, the vehicles and the infostations are the agents (actors) of the simulation, and they interact with other vehicles (transmit contents, request messages, etc.) or interact with the environment (in the city there are buildings that block communications if two vehicles are trying to share information and buildings are between them). The advantage of this agent based simulator is that it is a general purposes ones, so it does not restrict the programmer to do things in any particular way. MASON is a strong and efficient tool to implement complex individual and heterogeneous behaviors of entities that interact among themselves, which in platforms as OMNET++ (specific software for communications) can be hard to achieve. For example, for studying an incentive system to guarantee the cooperation of vehicles in forwarding traffic in the VANET, we need to develop different behaviors for vehicles. Some of them will be cooperative, some selfish, and some other malicious, each of them with different strategies to implement their behaviors. MASON is the right tool for this kind of studies. Therefore, the situation platform developed in this project is a good starting point for future research in those lines.

As already mentioned, three different protocols for content distribution have been implemented to perform the simulations, each one with a version with some optimizations. In each protocol, vehicles are looking for their desired content from the moment they enter to the simulation. They are trying to get the content through the VANET during 120 seconds, and if they can't get the content by the deadline, they download the content using their 3G/4G connection. Each vehicle has a 200 meters coverage radius in their transmissions, so vehicles further than 200 meters can't receive its messages. If two cars cannot see each other because there is a building in the middle, they will not be able to establish a communication either.

In the Push protocol, vehicles or infostations transmit on broadcast one random content they have stored in their cache every 10 seconds. So if one neighbor is interested on it, it can use it. An optimization to avoid that nearby vehicles transmit the same content within a short time, which is useless, is to add a time stamp to each content, so when a vehicle listen that one neighbor is transmitting that content, it puts the time stamp to 0, so this content will be the last to be chosen to be broadcast. Furthermore, in the optimized Push protocol version, when a content is downloaded using 3G network, this content is broadcast, because if it has been downloaded using the cellular network it implies that that content is not in this geographical area.

The Pull protocol transmits contents on demand, so each 10 seconds vehicles send a request message to ask neighbors for their desired content. If one of these neighbors has the content, it replies with it in unicast. When a vehicle gets its desired content, it stops sending request messages, but it is still traveling and sharing its content if someone asks for it. In the optimized version of the protocol, contents are sent in broadcast, so every neighbor can obtain the content hearing to the transmission. Moreover, when one content is downloaded using the 3G connection, it is also broadcast to every neighbor.

The MHTI protocol works in a similar way to the Pull protocol, but request messages are forwarded using geographical routing. If no neighbors have the requested content, they send their coordinates to the transmitter, so he chooses which one is closer to an infostation. This chosen vehicle asks for the same content, and if no one has it, it chooses again the closest vehicle to the target infostation, and so on until the request reaches the infostation or some vehicle in the path has the content. In the MHTI optimized protocol contents are sent in broadcast, and they are also broadcast when they have been downloaded using 3G.

Once we implemented these protocols, we have done some simulations varying traffic density, number of contents and popularity level. To measure the performance of each protocol, we chose as success rate (the proportion of contest obtained using the VANET). We have also measure the overhead injected to the network (ratio between bytes transmitted and bytes of contents received successfully). Finally, we have measured the number of contents stored in cache by each vehicle to check the spread rate.

The first studied situation is how each protocol behaves with only one content in the network and different traffic densities. We varied the number of vehicles from 2000 to 9000. Results have shown that every protocol get a good success rate (more than 90%) even in scenarios with low traffic density (2000 vehicles). Overhead is much bigger in Push protocols, because they are transmitting contents every 10 seconds while they are transmitting through the network, and Pull and MHTI only transmit contents on demand.

Then, the study moved to analyse number of contents fixing traffic density (to 6000 vehicles). Using only one content produces good results, but when number of contents is increased, it is not that easy to get the desired content. The number of contents considered goes from 1 content to 100.000. Push protocols' success rate falls to 0 when contents are above 100, so their overhead tends to infinity. Pull protocols produces better results, but success rate is about 20% when contents are above 100 contents. This success rate corresponds with vehicles that has found one infostation during their trip. MHTI achieves a reasonably good success rate even with 100.000 contents (55%), which implies that more than a half of the contents are downloaded using the VANET because requests reach infostations in a multi-hop process.

The next step has been to add popularity to the scenario, in a situation with 6000 vehicles and 1000 contents. Popularity tends to compensate the success rate falls with a large number of contents, but in Push protocols when the popularity is at his highest level (a Zipf distribution with alpha equals to 2), success rate in optimized Push is around 50%. Pull protocols also need a high level of popularity to achieve a performance over 60%, specifically the optimized version needs an alpha of 1.25. MHTI achieve a success rate of 80% in its optimized version with alpha equals to 1.25, reaching 90% with an alpha of 2.

The last studied situation considered a scenario similar to Internet, with 100.000 contents and a popularity level of alpha equal to 0.8, changing the traffic density. In this situation it has been demonstrated that MHTI is the only protocol that works when the users

demand is similar to the one in Internet. Push protocols has a 0% success rate and Pull stays at 20% no matter the traffic total traffic will be less density. MHTI starts with a 30% success rate with 2000 vehicles but reach 80% with 9000, what implies that in the situation in which the cellular network would be overloaded (high density of vehicles), the VANET will absorb most traffic, while in scenarios with a low density, and the role of the VANET would not be so critical.

Finally, results obtained in this work are quite similar to [9], which was at main objective task of this project. Our simulation platform has implemented this content distribution mechanism using an agent based model, a model that can deal with complex systems as content distribution with an incentive mechanism, which could not be easily developed in a generic simulator through for communications.